Extraction and Representation of Clinical Terms from Portuguese Clinical Text

Veniamin Craciun

Thesis to obtain the Master of Science Degree in
Information Systems and Computer Engineering

Supervisors: Prof. Bruno Emanuel da Graça Martins
Prof. Manuel Fernando Cabido Peres Lopes

Examination Committee
Chairperson: Prof. José Carlos Martins Delgado
Supervisor: Prof. Manuel Fernando Cabido Peres Lopes
Member of the Committee: Prof. Pável Pereira Calado

November 2019
Acknowledgments

I would like to thank both my supervisors Prof. Bruno Martins and Prof. Manuel Lopes for presenting me with the opportunity to work on a topic of my interest and for their patience, availability, guidance and support throughout the development of this thesis. Thank you!
Abstract

The adoption of electronic health records (EHRs) in clinics and hospitals made possible to aid healthcare professionals on a daily basis allowing to take decisions faster and better. However, EHRs often consist of large quantities of unstructured data, such as medical notes in free-text, which given their nature, contain several medical conditions, symptoms or clinical terms which could or could not be related in the same medical note, hence leading to ambiguity of which are the most important terms and how these relate to each other. This work proposes an architecture capable of processing text contained in clinical documents, extracting the key-concepts contained in them. This is accomplished through application of dimensionality reduction techniques, that transform the original high-dimensional data to a latent representation extracting only the most important features, such as non-negative matrix factorization and deep neural networks. After extraction and latent representation of key-concepts, is possible to analyze which are the most similar and how these relate to each other.

Keywords: clinical notes, clinical terms extraction, dimensionality reduction, deep neural networks, similarity
Resumo

A adopção de registos de saúde electrónicos (RSEs) em clínicas e hospitais permitem auxiliar os profissionais de saúde no seu dia-a-dia, possibilitando a tomada de melhores decisões e de forma mais rápida. Contudo, frequentemente os RSEs contêm dados sobre a forma não estruturada, tais como notas clínicas em texto livre, que dado a sua natureza, contêm múltiplas condições de saúde, sintomas, ou termos clínicos numa mesma nota, que podem ser ou não ser relacionados entre si, por isso criando ambiguidade sobre quais são os conceitos-chave e sobre que forma estão relacionados. Este trabalho propõe uma arquitectura capaz de processar texto contido em documentos clínicos de modo a extrair os conceitos-chave presentes nos mesmos. Isto é alcançado através da aplicação de técnicas de redução de dimensionalidade, que transformam dados altamente dimensionais para representações latentes extraindo apenas as características mais importantes, tais como factorização de matrizes não negativas ou redes neurais profundas. Após a extração e representação latente dos conceitos-chave, é possível analisar quais são os mais similares e conhecer como estes conceitos-chave se relacionam uns com outros.

Palavras-chave: notas clínicas, extração de termos clínicos, redução de dimensionalidade, redes neurais profundas, similaridade
4 Experimental Evaluation
  4.1 Datasets and Evaluation Methodology ........................................ 36
    4.1.1 Dataset .............................................................................. 36
    4.1.2 Dataset Exploration ............................................................. 37
    4.1.3 Dataset Transformation ....................................................... 38
    4.1.4 Evaluation Methodology ...................................................... 42
  4.2 Experimental Results .............................................................. 43
    4.2.1 Similarity between Clinical Terms ........................................ 43
    4.2.2 Quality of the Learned Representations ............................... 48

5 Conclusions and Future Work .................................................. 52
  5.1 Contributions ........................................................................... 52
  5.2 Future Work ............................................................................ 53

Bibliography ................................................................................. 54

A Code and Clinical Terms Similarity ....................................... 59

B Generated Topics ...................................................................... 71
List of Tables

4.1 Dataset structure before transformation .............................................. 37
4.2 Statistical characterization of the dataset ............................................. 37
4.3 Mapping snippet between ICD-10 and ICPC-2 ..................................... 40
4.4 Dataset structure after ICPC-2 transformation ..................................... 40
4.5 Average cosine similarity between pairs of codes ................................. 44
4.6 Average cosine similarity between pairs of phrases ............................... 44
4.7 Top ten features in five topics generated by FR-NMF model .................. 49
4.8 Average $k$-means clustering between representations .......................... 51

B.1 Topics generated by FR-NMF model. ................................................. 71
B.2 Topics generated by KL-NMF model. ................................................ 72
B.3 Topics generated by SNMF model. .................................................... 72
B.4 Topics generated by DSNMF model. .................................................. 72
B.5 Topics generated by AE model. ......................................................... 72
List of Figures

2.1 Word2Vec architecture overview .............................................. 11
2.2 PV-DM architecture overview ................................................ 12
2.3 Comparison between SNMF and DSNMF .................................... 14
2.4 Autoencoder architecture overview ......................................... 15
2.5 Variational Autoencoder architecture overview ......................... 17
2.6 Med2Vec architecture overview .............................................. 27
3.1 Outline of the proposed architecture ...................................... 33
4.1 ICD-10 code distribution in the dataset ................................. 38
4.2 ICD-10 chapter and block distribution ..................................... 39
4.3 Word cloud of the most common terms in the dataset .................. 39
4.4 ICPC-2 code distribution in the dataset .................................. 41
4.5 ICPC-2 chapter distribution in the dataset ............................... 41
4.6 Cosine similarity between pairs of codes produced by FR-NMF ... 45
4.7 Cosine similarity between pairs of key-phrases produced by FR-NMF 46
4.8 Cosine similarity between codes extracted by FR-NMF ............... 47
4.9 CV coherence results between models ...................................... 49
A.1 Similarity between codes from chapters 1 to 6 of ICD-10 by FR-NMF. 60
A.2 Similarity between codes from chapters from 7 to 12 of ICD-10 by FR-NMF model. . . 61
A.3 Similarity between clinical terms encoded with codes from chapters 1 to 6 of ICD-10 by FR-NMF .................................................. 62
A.4 Similarity between clinical terms encoded with codes from chapters 7 to 10 of ICD-10 by FR-NMF .................................................. 63
A.5 Similarity between clinical terms encoded with codes from chapters 11 to 20 of ICD-10 . 64
A.6 Similarity between clinical terms coded with the ICPC-2 classification by FR-NMF . . 65
A.7 Similarity between 30 most common ICD-10 codes in FR-NMF model. .................. 66
A.8 Similarity between 30 most common ICD-10 codes in KL-NMF model. .................. 67
A.9 Similarity between 30 most common ICD-10 codes in SNMF model. .................. 68
A.10 Similarity between 30 most common ICD-10 codes in DSNMF model. .................. 69
A.11 Similarity between 30 most common ICD-10 codes in AE model. .................. 70
Acronyms

AE  Autoencoder
DL  Deep Learning
DSNMF  Deep Semi Non-negative Matrix Factorization
EHR  Electronic Health Record
ICD  International Classification of Diseases
ICPC  International Classification of Primary Care
KL  Kullback-Leibler
LDA  Latent Dirichlet Allocation
LSA  Latent Semantic Analysis
NLP  Natural Language Processing
NMF  Non-negative Matrix Factorization
NMI  Normalized Mutual Information
NNDSVD  Nonnegative Double Singular Value Decomposition
PMI  Pointwise Mutual Information
SNMF  Semi Non-negative Matrix Factorization
SVD  Singular Value Decomposition
TF-IDF  Term Frequency - Inverse Document Frequency
UMLS  Unified Medical Language System
VAE  Variational Autoencoder
VSM  Vector Space Model
Chapter 1

Introduction

The introduction of Electronic Health Record (EHR) in clinics and hospitals, made possible to store and to aggregate clinical information from patients containing structured and unstructured data. For instance, the textual description written by a healthcare professional of a patient's reason for encounter (RFE) or complaint, i.e., a concise statement describing the symptom or problem is considered unstructured data. The challenges in working with unstructured data from an EHR system include high-dimensionality and high-variability (includes a wide range of data such as diagnosis, medication, test results), temporality which refers to the sequential nature of clinical events, limited interpretability due to the complexity of medical data, bias, systematic errors in the medical data, and mixed data types and missing data [Sadati et al., 2019].

Moreover, other challenges are related to those found in other Natural Language Processing (NLP) problems when dealing with human generated free-text, namely abbreviations, orthographic errors and ambiguities. In the NLP field these problems have been studied and can be tackled, to some degree, by employing word embeddings, also been used to extract medical terms [Vine et al., 2015] and to encode medical terms to compute similarities between patients [Choi et al., 2016, Gencoglu, 2019, Zhang et al., 2019, Zhu et al., 2016]. Meanwhile, other teams [Bhattacharya et al., 2016, Gefen et al., 2018] employed topic modelling methods, such as Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA), to find co-occurrences of health conditions using simpler vector space representations of the textual data.

Taking inspiration from previous teams, this work also studies the effect of vector representations and word embeddings combined with topic modelling to extract and to represent textual information from the medical domain.

1.1 Thesis Statement

The motivation behind the research proposal that is advanced in this document is to analyze medical records for extracting meaningful relationships contained on them, by applying representation learning techniques based on Non-negative Matrix Factorization (NMF) and its deep neural networks extensions.
Previous work has focused mainly on LSA and LDA models, giving little focus to the NMF approach and recent extensions leveraging Deep Learning (DL) advances. These NMF extensions have produced state-of-the-art results in clustering human faces [Trigeorgis et al., 2014], text classification [Flenner and Hunter, 2017] and topic extraction in short-text documents [Shi et al., 2018]. O’Callaghan et al. [2015] compared the use of NMF and LDA side-by-side under multiple datasets. The results show that LDA captures broader topics, while NMF produces more coherent and less generalized topics. Hence, the authors conclude that NMF is more suitable for analyzing niche or non-mainstream content. This suggests that NMF is a viable alternative to analyze textual information contained on medical records. For example, by applying NMF we expect to find if the descriptions given by health professionals differ semantically from the terminology used in the International Classification of Diseases (ICD\(^1\)) for code assigned. A NMF model can be used to extract topics from medical records, allowing us to group together unrelated medical terms and words used by health professionals.

In this line of work, we also propose to compare the hierarchical structure of two classification systems, namely International Classification of Diseases 10\(^{th}\) revision (ICD-10) and 2\(^{nd}\) version of International Classification of Primary Care (ICPC-2)\(^2\) by analyzing the underlying free-text descriptions and respective labels (i.e., ICD-10 and ICPC-2 codes assigned) given by healthcare professionals. To achieve this, several dimensionality reduction techniques were in place, based on both matrix factorization and deep neural networks, specifically methods based on NMF and autoencoder, respectively.

These methods allow us to create dense representation of the data, which not only removes noise but also enables the comparison of the learned representation against one another. For instance, allows to compare if latent representation of codes and key-phrases (i.e., clinical terms such as health conditions or symptoms) from the same chapter are more similar than those from different chapters. If that is the case it indicates that the descriptions and codes assigned by healthcare professionals follow the hierarchical structure of the classification system. Moreover, it also indicates that the classification system have a well-organized structure with same family of conditions aggregated on the same chapter and not distributed across several chapters. By applying the methods based on NMF and autoencoders, we expect to find if the descriptions given by healthcare professionals differ semantically from the terminology used in ICD and ICPC. Thus, this work aims at answering the following research questions: RQ-I) Which technique best preserves the hierarchical structure of ICD-10 and ICPC-2?, i.e., if codes and clinical terms from the same chapter show higher similarity than those from different chapters. RQ-II) Which dimensionality reduction technique performs the best from a topic modelling and clustering perspective?, i.e., verify which technique produce the representations that better aggregates related clinical terms and documents.

\(^1\)https://www.who.int/classifications/icd/en/
\(^2\)https://www.who.int/classifications/icd/adaptations/icpc2/en/
1.2 Methodology

The main objective of this proposal is to create an architecture that incorporates key-extraction and representation learning techniques to process textual information from a dataset composed of death certificates, written by a health professional and encoded with the ICD-10 classification. The challenges expected in this work are closely related to those found in other Natural Language Processing (NLP) problems, because the architecture needs to deal with human generated free-text. Hence, the architecture needs employ techniques proposed in the NLP and text mining fields to process these medical documents. But before starting the actual implementation, a great effort is given to understanding how each death certificate is composed, exploring which are the most frequent codes or and terms to describe the cause of death, thus assessing the structure as well as the content. This exploratory phase is critical, understanding structure and content of the data helps one to take precautionous measures that enhances the quality of the data given as input to the architecture. Furthermore, a special effort was given to get more knowledgeable with ICD-10 and ICPC-2 terminologies in order to understand how these are structured, thus be able to evaluate if the methods employed are producing coherent results according to structure of these terminologies, since it is one of the questions (RQ-I) of this work.

Only after the exploratory stage of the data and the objectives and challenges that the architecture needs to solve have been defined, the actual implementation of the a architecture commenced. In this stage surged the need to select the toolkit to solve the task at hand, resorting to Python as main programming language due to its affluence in the machine learning community. The toolkit was constituted by open-source libraries, such as Pandas\textsuperscript{3} and NumPy\textsuperscript{4} for data manipulation, scikit-learn\textsuperscript{5} for data representation and clustering evaluation, while Tensorflow\textsuperscript{6} was used to build neural networks models. Moreover, the toolkit also included Jupyter\textsuperscript{7}, matplotlib\textsuperscript{8} and seaborn\textsuperscript{9} for data exploration and visualization and Gensim\textsuperscript{10} library was employed to build document vector representations and to perform the topic modelling evaluation.

1.3 Results and Contributions

The main contributions of this work is the implementation of an architecture using the aforementioned toolkit which processes textual descriptions from clinical documents then through dimensionality reduction techniques it learns the latent representation of these documents. The learned representations are then used to extract the relevant clinical terms embedded in the documents and to compute the relationship between them. The performance of the architecture was evaluated based on cosine similarity between pairs of encoded clinical terms, measuring their ability to preserve hierarchical structure of two

\textsuperscript{3}https://pandas.pydata.org/
\textsuperscript{4}https://numpy.org/
\textsuperscript{5}https://scikit-learn.org/
\textsuperscript{6}https://www.tensorflow.org/
\textsuperscript{7}https://jupyter.org/
\textsuperscript{8}https://matplotlib.org/
\textsuperscript{9}https://seaborn.pydata.org/
\textsuperscript{10}https://radimrehurek.com/gensim/
classification systems, answering RQ-I, and based on topic modelling and clustering evaluation metrics, answering RQ-II. Moreover, the architecture allows to analyze the codes/phrases extracted by showing how these relate to each other. This is achieved by ranking and aggregating similar codes or phrases into topics or by computing their similarity in the latent space, thus allowing for easy visualization of how these are related.

1.4 Document Overview

This document has the following organization.

- Chapter 2 describes the theoretical concepts and related work, starts with the introduction and overview of two medical classification systems in Section 2.1, succeeded by a summary on representation of textual information, dimensionality reduction techniques and neural networks in Section 2.2, while Section 2.3 gives a bibliographic review of methods proposed to represent textual information in the medical domain powered by techniques proposed in the NLP field.

- Chapter 3 presents the technicalities of the proposed approach, introducing the architecture developed for extraction and representation of clinical terms through methods based on matrix factorization and autoencoders.

- Chapter 4 presents the experimental evaluation of the proposed architecture. Section 4.1 showcases the pre-processing of the data and gives a statistical analysis of the dataset used in the experiments, together with the experimental methodology and evaluation metrics. Section 4.2 discloses the results obtained during the experiments by the methods presented in Chapter 3.

- Finally, Chapter 5 outlines the main contributions and limitations of this work also providing possible developments for future work.
Chapter 2

Fundamental Concepts and Related Work

This chapter starts by accommodating the reader with standardized medical classification systems currently applied worldwide that are used by healthcare professionals to classify diseases, symptoms or medical procedures in Section 2.1. Then diving more deeply into the theory used in text mining and natural language processing fields to represent textual information in Section 2.2.1, to transform these representation into a latent space in Sections 2.2.2-2.2.4, to extract relevant information from text in Section 2.2.5 and introducing techniques proposed in these fields to measure the quality of latent representations in Section 2.2.6.

2.1 Medical Classification Systems

Medical classification systems are standardized systems used to aid the exchange of EHR between institutions or countries. Medical classification systems can be further divided into two subsystems, namely, diagnostic coding systems and procedural coding systems. Diagnostic coding systems are used to classify diseases, disorders, injuries and other related health conditions, examples of such systems are ICD 9th revision (ICD-9), ICD 10th revision (ICD-10) and SNOMED-CT. Procedural coding serve to indicate health interventions taken by healthcare professionals, example of such systems is the International Classification of Health Interventions (ICHI\(^1\)) and International Classification of Primary Care (ICPC\(^2\)).

Notably, ICD-10 has been proposed by World Health Organization (WHO) as a international standard that aims at closing the gap between existing coding systems proposed by several countries. ICD-10 is a hierarchical system currently adopted by more than 100 countries and translated into 43 languages, it consists of 22 chapters, where each chapter contains multiple blocks. In spite of the fact that ICD-10 is an international standard, several countries created modified version of ICD-10. To solve this issue, recently

\(^1\)https://www.who.int/classifications/ichi/en/
\(^2\)https://www.who.int/classifications/icd/adaptations/icpc2/en/
a new revision ICD-11 has been proposed. This revision aims at updating ICD-10 to the information age and allows to translate codes from and to ICD-10, is more flexible to be translated to more languages and structured to allow users to search the terms more easily. The new version is endorsed by all Member States of WHO with the estimated employed date, starting on January 2022.

Another widely used classification system is the ICPC-2, proposed World Organization of Family Doctors (WONCA) and accepted by WHO, has a biaxial structure and consists of 17 chapters, each divided into 7 components dealing with i) symptoms and complaints, ii) diagnostic, screening and preventive procedures, iii) medication, treatment and procedures, iv) test results, v) administrative, vi) referrals and other reasons for encounter and vii) diseases. ICPC-2 is structured differently than ICD-10, classifying patient data and clinical activity in the domains of general/family practice and primary care, taking into account the frequency distribution of problems seen in these domains. It allows classification of the patient’s reason for encounter (RFE), the problems/diagnosis managed, interventions, and the ordering of these data in an episode of care.

Both of classification systems are adopted in Portugal and currently in use in the vast majority of primary healthcare institutions to encode health complications. Although there is a possible mapping between the two classification systems, these have been used in different contexts. For instance, ICPC-2 is commonly used by family doctors because this classification system is more generalist and closer to the terms used by patients to describe symptoms or reason for encounter upon visiting a health clinic/center. Meanwhile, ICD-10 is highly specialized on encoding clearly defined health problems and more thoroughly used in hospitals. Hence, by comparing the two classification systems one is able to find the what are the main differences regarding the clinical terms used and how these are related.

2.2 Fundamental Concepts

This chapter gives an overview of the underlying theory regarding the methods used throughout the implementation, with special emphasis on the representation of textual information to learn and extract features present on the data while at the same time to condense the original information contained in it.

2.2.1 The Vector Space Model for Representing Textual Documents

The Vector Space Model (VSM) is a general approach for representing a set of documents $D$, over a vocabulary of words $V$, as a matrix $C$ where each cell $[j,i]$ encodes the relation between document $j \in D$ and vocabulary term $i \in V$. The index terms can refer to individual words, $n$-grams (i.e., sequences of $n$ continuous words), or character $n$-grams. The association scores can for instance correspond to the number of occurrences of a word $i$ in document $j$, to a TF-IDF score computed as shown in 2.1, or any other function that scores the association of words to documents.

\[
TF(t,d) = \frac{w_{td}}{\sum_k w_{kd}}, \quad IDF(t) = \log \left( \frac{|D|}{n_t} \right)
\]

\[
TFIDF(t,d) = TF(t,d) \times IDF(t)
\]

(2.1)
In the previous equation, \( w_{td} \) is the frequency of the term \( t \in V \) in document \( d \in D \), while \( n_t \) is the number of documents from the collection \( D \) in which term \( t \) occurs. This simple and intuitive representation allows one to compute similarity between word vectors (rows) or between document vectors (columns), using a similarity function such as cosine similarity. One can compute the cosine similarity between vectors \( x \) and \( y \) using 2.2, which corresponds to calculating the cosine of the angle between two vectors. The similarity between word vectors (or documents) is inversely proportional to angle between them, i.e., a smaller the angle between \( x \) and \( y \) implies that they are similar.

\[
\cos(x, y) = \frac{x \cdot y}{\|x\| \times \|y\|}
\]

(2.2)

Although simple and effective, the VSM also has some limitations when encoding large document collections, or when considering large vocabulary sizes. In such cases, the VSM representations are high-dimensional sparse matrices, meaning that the most of the cells in the matrix are zero, difficulting to do comparison operations between documents. The dimensionality of a matrix can be reduced to a more compact representation using multiple techniques and several extensions have been proposed leveraging matrix factorization techniques, as will be discussed in Section 2.2.2.

Additionally, a second drawback of standard VSM representations lies in the fact that these are incapable of capturing the semantics of words within documents, thus having problems in dealing with synonyms and polysemous words. This problem can be tackled, to some degree, by leveraging statistical language models, such as embeddings.

### 2.2.2 Dimensionality Reduction Techniques

The next sections describe two approaches widely used for dimensionality reduction, which are also the basis for other techniques and extensions that address the problem of high-dimensionality and sparsity previously introduced by VSM, namely singular-value decomposition and non-negative matrix factorization.

**Singular-Value Decomposition**

A widely used approach for dimensionality reduction is Singular Value Decomposition (SVD). It consists on decomposing a matrix \( C \) into the multiplication of three different matrices, as shown below:

\[
C = U_0 \cdot \Sigma_0 \cdot V_0^T
\]

(2.3)

In the previous expression, \( U_0 \in \mathbb{R}^{v \times m} \), \( V_0 \in \mathbb{R}^{m \times d} \) and \( \Sigma_0 \in \mathbb{R}^{m \times m} \) is a matrix where the diagonal contains positive values in decreasing order, while the remaining entries are zero. The parameters \( v \) and \( d \) represent respectively the size of vocabulary and documents, these parameters will be used throughout this document with the same meaning, while \( m \) represents the rank of the matrix \( \Sigma_0 \), and consecutively of \( C \). The diagonal values (\( \sigma_i \)) are ordered based on their importance, hence the resulting matrices can be approximated creating smaller matrices.
This can be achieved by considering only the top \( k \), with \( k \leq m \), values on matrix \( \Sigma_0 \), zeroing the remaining diagonal entries, and this way producing a new matrix \( \Sigma \). Since all entries greater than \( k \) are zero, then the corresponding columns of \( U_0 \) and \( V_0 \) are deleted, obtaining \( U \) and \( V \). The product of the resulting matrices, \( \hat{C} \), is only approximately equal to \( C \). This method is called best rank-\( k \) approximation and can be used to represent \( C \) as a projection to a lower rank matrix (i.e., components are truncated into a lower dimensionality, similarly to what happens when applying principal component analysis).

Equation 2.4 displays a visual representation for \( \hat{C} \).

\[
\hat{C} = U \cdot \Sigma \cdot V^T = \begin{bmatrix} u_1 & u_2 & \ldots & u_k \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_k^T \end{bmatrix} (2.4)
\]

Notice that \( U \) and \( V \) are orthonormal, i.e., their rows are both unit-length vectors and orthogonal to each other. The LSA model Deerwester et al. [1990] builds upon SVD and uses the resulting matrices to compute the similarity between words or documents. In order to compute similarities between words, consider the word vectors matrix:

\[
W = U \cdot \Sigma, \quad W \in \mathbb{R}^{v \times k} \text{ and } W_i \in \mathbb{R}^k (2.5)
\]

In the previous expression, the row \( W_i \) correspond to word \( i \) in the vocabulary. The similarity between the words \( i \) and \( j \) can be computed, using Equation 2.2, as \( \cos(W_i, W_j) \). Likewise, to compute document similarity, we can consider:

\[
D = \Sigma \cdot V, \quad D \in \mathbb{R}^{k \times d} \text{ and } D_i \in \mathbb{R}^k (2.6)
\]

In the previous expression, the column \( D_i \) correspond to document \( i \) in the corpus. Again the similarity between documents can be computed as \( \cos(D_{i1}, D_{j1}) \).

The performance of the LSA model depends on the parameter \( k \). Deerwester et al. [1990] argue that the selection of \( k \) is crucial because the value for \( k \) should be large enough to fit the real structure of the data, but small enough so that the model does not fit noise and unimportant details. The authors suggest to execute several runs with different values for \( k \), and then use the one that yields the best performance in the envisioned downstream task.

**Non-Negative Matrix Factorization**

Another dimensionality reduction technique is Non-negative Matrix Factorization (NMF). This approach is very similar to SVD but, in this case, the matrix \( C \) is decomposed into the multiplication of two non-negative matrices, as shown in Equation 2.7.
\[ C \approx W \cdot H = \begin{bmatrix} w_1 & w_2 & \ldots & w_k \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix}, \text{ with } w_i \in \mathbb{R}_+^d \text{ and } h_i \in \mathbb{R}_+^v \] (2.7)

The approximation can be formulated as an optimization problem that aims at minimizing the reconstruction error between the multiplication of the factorized matrices, \( W \cdot H \), and the original matrix \( C \). Several algorithms have been proposed to solve the optimization problem, but only two algorithms will be described here, following the presentation by Lee and Seung [2000]. In the first optimization problem, corresponding to Equation 2.8, the algorithm consists on minimizing the Frobenius norm (or distance) between the original matrix \( C \) and the resulting multiplication of two non-negative matrices \( W \) and \( H \).

\[
\min_{W,H} \|C - WH\|_F^2 = \sum_{ij} \left( C_{ij} - (W \cdot H)_{ij} \right)^2, \text{ where } W,H \geq 0 \tag{2.8}
\]

The second optimization problem considers a modified version of the Kullback-Leibler (KL) divergence (or relative entropy), formalized as follows:\n
\[
\min_{W,H} \sum_{ij} \left( C_{ij} \log \left( \frac{C_{ij}}{(W \cdot H)_{ij}} \right) - C_{ij} + (W \cdot H)_{ij} \right), \text{ where } W,H \geq 0 \tag{2.9}
\]

The previous equation reduces to original Kullback-Leibler formula when the following condition holds (i.e., when both \( C \) and \( W \cdot H \) are normalized distributions):

\[
\sum_{ij} C_{ij} = \sum_{ij} (W \cdot H)_{ij} = 1 \tag{2.10}
\]

Regardless of the chosen optimization problem, both can be solved using multiplicative updates or gradient descent (i.e., additive updates). However, the gradient descent approach is step-size sensitive and thus choosing a adequate step-size is crucial for finding a low reconstruction error. A small value for the step-size increases the iterations needed to find a minima, while a large step-size might produce jumps without converging to a local minima. Lee and Seung [2000] argue that multiplicative updates are well-suited for the majority of tasks, having a good compromise between computational overhead and ease of implementation. Equation 2.11 shows the multiplicative update rules that are used to update both \( H \) and \( W \) when considering the Frobenius norm.

\[
H_{\alpha \mu} = H_{\alpha \mu} \frac{(W^T \cdot C)_{\alpha \mu}}{(W^T \cdot W \cdot H)_{\alpha \mu}} \quad W_{\alpha a} = W_{\alpha a} \frac{(C \cdot H^T)_{\alpha a}}{(W \cdot H \cdot H^T)_{\alpha a}} \tag{2.11}
\]

Similarly, Equation 2.12 shows multiplicative update rules when using the KL divergence to solve the optimization problem.

\[
H_{\alpha \mu} = H_{\alpha \mu} \frac{\sum_i \left( W_{\alpha a} \cdot C_{i \mu} / (W \cdot H)_{i \mu} \right)}{\sum_k W_{k a}} \quad W_{\alpha a} = W_{\alpha a} \frac{\sum_i \left( H_{\alpha \mu} \cdot C_{i \mu} / (W \cdot H)_{i \mu} \right)}{\sum_v H_{v a}} \tag{2.12}
\]
It is assumed that both the Frobenius distance and the KL divergence between \( C \) and \( W \cdot H \), under multiplicative updates, is non-increasing hence the multiplicative factor that updates \( H \) and \( W \) is unity (i.e., zero error reconstruction) when:

\[
C = W \cdot H
\]  

(2.13)

In order to reduce the number of steps needed for convergence, while maintaining a low reconstruction error, different approaches have been suggested in the literature as the initialization of matrices \( W \) and \( H \), aside from considering a random initialization (i.e., initializing both matrices with random positive numbers). Most approaches consider a SVD-based initialization Boutsidis and Gallopoulos [2008], Qiao [2015] but other proposals include the use of \( k \)-means Wild et al. [2004], although the \( k \)-means approximation is too computationally expensive to be used in practice as a NMF initialization, for large problems.

Another approach is to run multiple random initialization and pick the best one. This approach not only is more expensive computationally but also does not produce unique \( W \) and \( H \) matrices. To tackle this issue, Boutsidis and Gallopoulos [2008] introduced a new initialization approach named NNDSVD that it is simple to adapt into current NMF algorithms, does not contain randomized values, reduces the reconstruction error, and is proved to converge faster. As the name suggests, this approach consists in decomposing matrix \( C \) into a lower rank-\( k \) matrix through SVD. Once the lower rank matrices \((U, \Sigma, V)\) are found the algorithm multiplies the columns of \( \Sigma \) and \( U \) to initialize the columns of \( W \) and multiplies the columns of \( \Sigma \) and \( V \) to initialize the columns of \( H \). Doing so, the matrices \( W \) and \( H \) are initialized using values from the data itself, guaranteeing convergence to a local minima.

**SemiNMF**

Several extensions have been proposed to the standard NMF algorithm. For instance, Semi Non-negative Matrix Factorization (SNMF) Ding et al. [2010] is a matrix factorization technique that learns a low-dimensional representation of a dataset that lends itself to a clustering interpretation. SNMF differs from NMF because the matrix \( W \) has no restriction on the values that it can take, i.e., \( W \in \mathbb{R}^{d \times k} \) while the restriction on \( H \) maintains to non-negative values, i.e., \( H \in \mathbb{R}^{k \times v}_+ \). The authors argue that SNMF helps to close the bridge between NMF and \( k \)-means clustering. In this case the matrix \( W \) serves as the cluster centroids while \( H \) can be viewed as the cluster indicators for each data-point.

### 2.2.3 Statistical Language Models

Language models are a important subfield of NLP, serving as the main building blocks for applications such as machine translation, question answering, spelling correction, etc. Specifically, statistical language models aim at estimating a probability distribution over linguistic units such as sentences, words or characters. The \( n \)-gram model is one of the simplest approaches, predicting the probability of seeing word \( w \) given a history of previous words \( h \), where the history contains \( n - 1 \) words. Nonetheless, the \( n \)-gram model can have problems in terms of capturing and encoding the semantics of language units, and thus newer approaches have been proposed that make use of neural networks to encode text. For
Figure 2.1: Architectural outline of Word2Vec models, as introduced by Mikolov et al. [2013]. The input words are projected to columns of the matrix \( M \) to predict the output word.

instance, word embedding techniques attempt to encode words as real valued vectors capturing semantic and syntactic information through optimization problems that leverage contextual information on word occurrences. The next sections give an overview on word embedding approaches in the field of NLP.

**Word2Vec**

One of the most notable contributions to the field of representation learning was made by Mikolov et al. [2013] with the introduction of Word2Vec. This framework proposes two different models; namely the contiguous bag of words (CBOW) and skip-gram models, to compute vector representations from words. The CBOW model aims at keeping the context of a word in a sentence, hence this model takes as input multiple vectors (i.e., multiple words) and outputs a single vector encoding a dense representation for the word in the middle of a sentence. For example, considering a sentence composed of five words and given the first and last two words, the CBOW model will predict the word that should be in the middle of the sentence. The skip-gram model is nearly the inverse process of CBOW, where a single input is mapped to various output vectors. For example, given any word as input the skip-gram approach will predict the surrounding words. Figure 2.1 illustrates the architecture of both the CBOW and skip-gram models, making it easier to understand the input and output of the models.

In Word2Vec, every word is mapped to a unique vector, represented by a column in a matrix \( M \). The column is indexed by position of the word in the vocabulary. In the CBOW model, the sum of the vectors is then used as features for prediction of a word in a sentence. Specifically, given a sequence of training words \( w_1, w_2, w_3, \ldots, w_T \), the objective of the model is maximize the average log probability:

\[
\frac{1}{T} \sum_{i=k}^{T-k} \log p(w_i | w_{i-k}, \ldots, w_{i+k}) \tag{2.14}
\]
where \( k \) is the context window size around every word \( w_i \). To predict the word in the context can be achieved via multiclass classifier such as softmax, defined by:

\[
p(w_t|w_{t-k}, \ldots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}
\]

(2.15)

where each \( y_i \) is un-normalized log-probability for each output word \( i \), computed as:

\[
y = b + Uh(w_{t-k}, \ldots, w_{t+k}; M)
\]

(2.16)

where \( U, b \) are the softmax parameters, and \( h \) is constructed by a concatenation or average of word vectors extracted from \( M \).

The neural network based word vectors are usually trained using stochastic gradient descent where the gradient is obtained via backpropagation. After the training converges, words with similar meaning are mapped to a similar position in the vector space. For instance, the words \( \text{car} \) and \( \text{automobile} \) will close to each other in the latent space as these appear in similar context, furthermore this representation also captures the implicit relationship between two terms, e.g., the name of a country and its capital city, such as \( \text{France} \) and \( \text{Paris} \) or \( \text{Italy} \) and \( \text{Rome} \).

**Doc2Vec**

In a similar line of work, Le and Mikolov [2014] introduced another word representation architecture named Paragraph Vector (more known in the literature as Doc2Vec) which builds upon then Word2Vec technique presented previously. This new technique allows to encode variable-length pieces of text from key-phrases, sentences or even documents. Similarly to Word2Vec, this technique also introduces two different models, namely Paragraph Vector Distributed Memory (PV-DM) sharing the traits of CBOW and Paragraph Vector Distributed Bags of Words (PV-DBOW) similar to skip-gram model.

In PV-DM every paragraph (or document) is mapped to a unique vector \( P_v \), represented by a column in matrix \( D \) and every word \( w_i \) is also mapped to a unique vector, represented by a column in matrix
The extra paragraph vector $P_t$ is used to encode the context in which the words appear, then is concatenated with several word vectors $(w_{t-k}, \ldots, w_{t-2}, w_{t-1}, w_t)$ in that context to predict the next word $w_{t+1}$, similar to the CBOW architecture as shown in Figure 2.2. The main difference in this model compared to the CBOW framework is in Equation 2.16, where $h$ is constructed from $M$ and $D$.

Meanwhile, in the PV-DBOW model the context words are ignored, instead the model is forced to predict the output by randomly sampling the words in the paragraph similarly to skip-gram model in Word2Vec, however instead of using a word-vector it uses a paragraph vector, and having that same training process as in Word2Vec. The authors show through experiments that using PV-DM is consistently better than PV-DBOW on text classification and sentiment analysis datasets. The authors recommend to combine both models in order to produce more consistent results across datasets.

### 2.2.4 Neural Networks for Representation Learning

This section starts by describing an extension to the SNMF method, specifically Deep Semi Non-negative Matrix Factorization (DSNMF), which is powered by deep neural networks to achieve multi-step factorization at every hidden network layer, and where each layer encodes different information. Moreover, this section gives an in-depth introduction to a sub-family of neural networks called autoencoders, which have introduced as unsupervised representation/feature learning techniques by reducing the dimensionality of the data similar to NMF method. This section also introduces and compares the traditional Autoencoder (AE) against recent extensions such as the Variational Autoencoder (VAE).

**DeepSemiNMF**

Trigeorgis et al. [2014] argue that the SNMF approach is not able to capture lower-level hidden attributes on a dataset that contains complex hierarchical information. Thus, the authors indicate that by factorizing the matrix $H$ in several steps, one is able to find these lower-level hidden attributes. The authors proposed a novel model, named DSNMF, that is able to learn hidden representations that allow themselves to clustering interpretation, according to different attributes of a given dataset.

The diagram in Figure 2.3 allows us to better understand the differences between the SNMF model and DSNMF. The former is a simple linear transformation of the initial input space, whereas the latter learns a hierarchy of hidden representations aiming to uncover the low-dimensional representation of the input. For instance, by employing the latter technique, one is able to find low-level features from a human face (e.g., facial expression or pose). This can be achieved by successively factorizing a representation matrix $C$ into $m$ steps, as follows:

$$C^\pm \approx W_1^\pm \cdot W_2^\pm \cdots W_m^\pm \cdot H_m^+$$

(2.17)

The notation used in the previous equation is similar to the notation used by Trigeorgis et al. [2014], and having $M^\pm$ it means that a matrix $M$ can take any real value, where $M^+$ can only take positive values.
Figure 2.3: Comparison between SNMF and DSNMF, depicting the multi-step factorization of matrix $C$ in DSNMF.

values. The previous equation can also be understood as if the SNMF has been applied successively over the matrix $H$ at step $m - 1$, resulting in matrices $W$ and $H$ at step $m$, defined by the following equations:

$$H_{m-1}^+ \approx W_m^+ \cdot H_m^+$$

$$\vdots$$

$$H^+_1 \approx W_2^+ \cdot H_2^+$$

The main difference of this model was the introduction of a non-linear function $g(\cdot)$ to approximate the original input matrix $C$ using $H$. Hence, $H_m$ is approximated, as follows:

$$H_m \approx g(W_{m+1} \cdot H_{m+1})$$

For training this model, the authors proposed to use a modified version of the Frobenius norm (Equation 2.8) as the optimization problem, minimizing the following equation:

$$\frac{1}{2} \| C - W_1 \cdot g(W_2 \cdot g(\ldots g(W_m \cdot H_m))) \|_F^2$$

After training, the authors measured the performance of the DSNMF model against the SNMF and NMF variants on three different datasets. Namely, the CMU MultiPIE [Gross et al., 2010], CMU PIE [Sim et al., 2003] and XM2VTS [Messer et al., 1999]. Each dataset contains a collection of images under different lightning conditions, from different human subjects (identity), expressing a variety of emotions (expressions) taken from many angles (pose). Each image served as the input matrix $C$. The authors show that by applying DSNMF, each matrix $H_m$ holds different features. For instance, for a DSNMF model with 3 hidden layers, the authors show that in the first layer, i.e., $H_1$ holds information regarding the pose of a face, whereas $H_2$ contains features regarding the facial expression, and $H_3$ represents the identity of a person. They also show, through examples, that their approach is able to outperform not only SNMF but also other NMF variants at learning low-dimensional representations for clustering purposes. During our experiments, the DSNMF model is compared against SNMF and NMF techniques.
on multiple clustering metrics to assess if this method does indeed perform better from a clustering perspective.

**Autoencoder**

The Autoencoder (AE) have been proposed by Hinton and Salakhutdinov [2006] as a new method for reducing the dimensionality of data in an unsupervised learning setting capable of retaining the important features present in the data. The autoencoder was proposed as a non-linear generalization of PCA and offered as powerful alternative, outperforming it in clustering tasks.

The architecture of an autoencoder is comprised of three main components:

- **Encoder**: consists of a series of layers with decreasing number of nodes and ultimately reduces to a latent space representation.

- **Latent Representation**: is lowest level space in which the inputs are reduced and information is preserved.

- **Decoder**: is the mirror image of the encoder but in which number of nodes in every layer increases and ultimately outputs the similar (almost) input.

The encoder is used to transform highly dimensional data into a smaller space, while the decoder is used to transform back the learned features from the latent space back to a reconstruction of the original space. Both encoder and decoder can contain one or several layers, forming deep neural networks.

The model contains an encoder function $g(\cdot)$ parameterized by $\phi$ and a decoder function $f(\cdot)$ parameterized by $\theta$. Given some input $x$, the low-dimensional latent representation that is learned is $z$ and the reconstructed input is $\hat{x} = f_\theta(g_\phi(x))$.

The parameters $(\theta, \phi)$ are learned together to output a reconstructed data sample same as the original input, $x \approx f_\theta(g_\phi(x))$, or in other words, to learn an identity function. There are various metrics
to quantify the difference between two vectors, one of the simplest methods is the Mean Square Error (MSE) loss, hence the loss function of the AE is shown in Equation 2.21.

\[
L_{AE}(\theta, \phi) = \frac{1}{2} \sum_{i=1}^{n} (x^{(i)} - f_{\theta}(g_{\phi}(x^{(i)})))^2
\] (2.21)

Since autoencoders learn a identify function, it tends to copy or overfit the input data, producing weak generalizations of the input data. More recently, multiple variants of autoencoders have been proposed to tackle this problem, such as Denoising autoencoder (DAE) by Vincent et al. [2008], Variational Autoencoder (VAE) by Kingma and Welling [2013] and more recently β-VAE by Higgins et al. [2017]. As the name suggests, DAE aim at removing the noise from the input, e.g., is able of reconstructing missing parts in a image, thus has been specifically crafted to be more robust against corrupted input. Meanwhile, VAE is a generative model, that allows to create new data, for example, is able to generate real looking human faces after the model was trained on a dataset of human faces. Both approaches create stronger generalizations of the input than traditional autoencoders.

**Variational Autoencoder**

The Variational Autoencoder (VAE) differs from traditional autoencoders because the learned latent representation is not a fixed vector but instead is a distribution. Let us assume that the data has distribution \( p_{\theta}(X) \), according to a particular set of points \( X = [x^{(1)}, ..., x^{(N)}] \). Typically, such a distribution can be represented as follows:

\[
p_{\theta}(X) = \prod_{n=1}^{N} p_{\theta}(x^{(i)})
\] (2.22)

Now let us consider that for each data point there is a latent variable \( z \), thus for a single data point the above becomes:

\[
p_{\theta}(x) = \int p_{\theta}(x, z) \, dz = \int p_{\theta}(z) \, p_{\theta}(x|z) \, dz
\] (2.23)

Let us also assume that there is a real parameter \( \theta^* \) that is given for this distribution, then one can generate a sample that assembles the original data point \( x^{(i)} \) following a generative process consisting of two steps:

- **Sample** \( z^{(i)} \sim p_{\theta^*}(z) \)
- **Generate** \( x^{(i)} \) given the conditional distribution \( p_{\theta^*}(x|z = z^{(i)}) \)

However, we do not know anything about the likelihood \( p_{\theta^*}(x|z) \) or the prior \( p_{\theta^*}(z) \), this can be solved by following Bayes’ rule, as follows:

\[
p_{\theta}(z|x) = \frac{p_{\theta}(x|z) \, p_{\theta}(z)}{p_{\theta}(x)}
\] (2.24)

Moreover, since we have to calculate \( p_{\theta}(x) \) using Equation 2.22 and it is high-dimensional thus difficult to compute, because we need to check for all possible values of \( z \), this problem is known as intractable inference. To solve this issue, an approximation \( q_{\phi}(z|x) \) can be used parametrised by \( \phi \) instead. Thus, the objective of VAE is very similar of that of NMF, however instead of minimizing the KL divergence
between two matrices, it minimizes the KL divergence between the two probability distributions, as follows:

$$\min D_{KL}(q_{\phi}(z|x)||p_{\theta}(z|x)), \quad w.r.t \quad \phi$$  \hspace{1cm} (2.25)

where $q_{\phi}(z|x)$ is the learned distribution. Doersch [2016] derives Equation 2.25 to obtain the loss function defined as:

$$\mathcal{L}_{VAE}(\theta, \phi) = -D_{KL}(q_{\phi}(z|x)||p_{\theta}(z|x)) + \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)]$$  \hspace{1cm} (2.26)

The first term constrains distribution $q_{\phi}(z|x)$ to take the shape of $p_{\theta}(z)$ which can be easily achieved using a Gaussian distribution, meanwhile the second term maximizes the reconstruction of input with a given latent variable $z$ given $p_{\phi}(x|z)$. In order to approximate VAE to AE one can consider the conditional probability $p_{\theta}(x|z)$, known as probabilistic decoder to be similar to the decoder $f_{\theta}(\cdot)$ introduced before, while the approximation function $q_{\phi}(z|x)$ is the probabilistic encoder, playing a similar role as $g_{\phi}(\cdot)$ in the traditional autoencoder.

Recently, Squires et al. [2019] and Burkhardt and Kramer [2019] introduced extensions to VAE similar to NMF and LDA, namely Probabilistic Autoencoder Non-negative Factorization (PAE-NMF) and Dirichlet Variational Autoencoder (DVAE). Both teams argue that these new methods allow for interpretable representations by restricting the latent layer to be non-negative, inheriting the same properties of NMF. These extensions differ from VAE by sampling from Weibull and Dirichlet distributions instead of Gaussian. Furthermore, Burkhardt and Kramer [2019] work builds on top of Neural Variational Document Model (NVDM) proposed by Miao et al. [2016]. NVDM uses an autoencoder architecture which was specifically crafted for text processing.

### 2.2.5 Key-phrase Extraction

This section gives an overview of multiple existing key-phrase extraction approaches. These approaches can be divided into two groups, i.e., unsupervised and supervised, the former can be fur-
ther divided into four subcategories, namely, statistics-based, graph-based, embeddings-based and language model-based, while latter are divided into two subcategories, namely traditional neural network approaches (e.g., binary classification and learning to rank) and deep learning. The following sub-sections covers in detail only a subset of statistics-based unsupervised techniques, although there are several other unsupervised and supervised techniques proposed that are not described here, as during the experiments were employed only statistics-based approaches.

Statistics-based Methods

One of the simplest and fast statistics-based baseline is TF-IDF, where each key-phrase (can be composed of either a single word or multiple words) has assigned a score and ranked according to the formula in Equation 2.1. However, more recently, several approaches have been proposed that take advantage of the co-occurrence statistics from external sources as is the case of AutoPhrase [Liu et al., 2015, Shang et al., 2018]. AutoPhrase allows to extract relevant information from documents given a knowledge base (i.e., terminological resources such as Wikipedia) available on the targeted language. AutoPhrase does not require human intervention to select quality phrases, because these are automatically selected from the provided knowledge base. The proposed approach is also domain-independent, aiming to enable automated phrase mining from different types of documents given as input. This is useful in tasks where human-knowledge is expensive or nonexistent.

In the AutoPhrase framework, in order to select quality phrases, the first step consists of building a candidate set of potential phrases based on their $n$-gram frequency, by imposing a threshold for frequency or by choosing a value for $n$. Once the candidate phrases are created, AutoPhrase uses the knowledge base to create two pools of phrases, namely a positive pool and a noisy negative pool (see Figure 3.1). The assignment of phrases to a pool depends on their quality and it uses four different criteria, described below:

- **Popularity**: quality phrases should occur with sufficient frequency in the given document collection, e.g., currently the word *database* is more popular than the original version *data base*.

- **Concordance**: the collocation of tokens present in quality phrases occurs with significantly higher probability than expected due to chance, e.g., the phrase *strong tea* is considered more concordant than *powerful tea* because the former is used more frequently.

- **Informativeness**: a phrase is informative if it is indicative of a specific topic or concept, e.g., *this paper* is a popular and concordant phrase, but does not add any additional information in a research publication corpus.

- **Completeness**: long frequent phrases, and subsequences from those phrases, may both satisfy the 3 criteria above. A phrase is deemed complete when it can be interpreted as a complete semantic unit in some given document context. Note that a phrase and a subphrase contained within it may both be deemed complete, depending on the contexts in which they appear. For
example, the phrases relational database system, relational database and database system can all be valid in certain contexts.

The four criteria explained above can be achieved by employing the following metrics. In order to select only popular phrases, one must impose a minimum threshold $\theta$ on the counts for all phrases, i.e., do not consider phrases that occur less than $\theta$. If a phrase $v$ is popular, then one can find the most concordant subsets $\langle u_i, u_j \rangle$ in $v$ by minimizing the PMI function, as presented in the next expressions:

$$P(u_i) = \text{TF}(u_i, v) \quad P(u_j) = \text{TF}(u_j, v)$$

$$\text{PMI}(u_i, u_j) = \log \left( \frac{P(v)}{P(u_i) \cdot P(u_j)} \right)$$

The informativeness of a phrase can be estimated by computing the average over all IDF weights $w_i$, for the words in $v$, i.e., each weight encodes how relevant a word is in a collection, hence the average over these weights allow to estimate how important a phrase is. Moreover, the authors also use the punctuation and capitalization hints to measure the informativeness of a phrase, e.g., by measuring the probability of a phrase occurring between quotes, a higher probability indicates that a phrase is informative. Based on the previous metrics, the positive and negative pools of phrases are generated, these can be the final output of the framework or they can be further processed using a part-of-speech (POS) tagger. In the latter case, the quality of each phrase is re-estimated by finding the best way of segmenting/splitting a document or sentence. The authors argue that POS tags support selecting better breakpoints in a sentence, i.e., it addresses the challenge of measuring completeness. Hence, because POS tags hold meta-information about a sentence, the previous example relational database system will be considered a complete phrase.

The architecture advanced in the current work employs AutoPhrase during key-extraction stage, because of the advantages that come with the four criteria previously explained. Another reason for choosing AutoPhrase as the framework for extracting key-phrase is that AutoPhrase learns the important features by using an external knowledge base since for our particular dataset it is an utmost requirement, because it contains medical terms and codes which are not commonly found in spoken language, hence a specialized knowledge base is needed in order to extract the terms related to the medical domain and achieve measurable results.

A recent survey by Papagiannopoulou and Tsoumakas [2019] explored multiple key-phrase extraction approaches on three different types of datasets, namely, the authors used full-text scientific articles, abstracts only from scientific articles and news. In case of key-phrase extraction from full-texts of scientific articles, statistical methods based on TF-IDF have yielded better results, offering a good balance between performance and computation time compared to the graph-based methods such as TextRank, proposed by Mihalcea and Tarau [2004], that achieve lower accuracy and have higher computation time. However, in case of key-phrase extraction from short texts, graph-based methods are suggested as more appropriate, since their computation cost is not considerable due to the short length of texts, and they perform better than the statistical methods. The reason being that in graph-based methods, the information that comes from the graph-of-words and co-occurrence statistics are sufficient. There is
no need for any additional positional or topical information, as the words do not appear in a long text (position does not play an important role in documents of a few lines) and the frequencies of similar key-phrase candidates are quite low due to the limited text.

The authors of the survey also suggest to consider both statistical and graph-based methods for datasets composed of text documents that are not from a specific domain, as these methods achieve better results in exact and partial matching of key-phrases. The survey also introduces recent methods that have been proposed which take advantage of both statistics and contexts information, such as YAKE Campos et al. [2019]. YAKE is unsupervised statistics-based method, i.e., it uses the term’s position/frequency, also uses new statistical metrics that capture context information and the spread of the terms into the document. YAKE can be understood as a framework being composed of five steps:

1. **Text pre-processing**: the given text corpus is split into sentences aided by punctuation clues and other delimiters, other noisy information is removed in this step.

2. **Feature extraction and term scoring**: uses a set of features to compute the score for each word in each sentence given by Equation 2.29.

3. **Candidate key-phrase generation and scoring**: using a \( n \)-gram sliding window, each sentence is split into chunks to create key-phrases, each key-phrase is assigned a score based on Equation 2.30.

4. **Data deduplication**: works by computing the distance between key-phrases then removing those that distance is lower than a threshold \( \theta \).

5. **Ranking of key-phrases**: all key-phrases are sorted by ascending score, a lower score assigned to a key-phrase is better.

After text pre-processing, during the feature extraction step, a set of five features are calculated for each individual term, these are:

- \( W_{case} \) - reflecting the casing aspect of a word.
- \( W_{position} \) - that values more the words occurring at the beginning of a document.
- \( W_{freq} \) - word frequency.
- \( W_{rel} \) - word relatedness to context that computes the number of different terms that occur to the left/right side of the candidate word.
- \( W_{sentence} \) - that quantifies how often a candidate word appears within different sentences.

Then, all these features are used for the computation of the \( S(w) \) score for each term (the smaller the score, the more important the word \( w \) is), given by the following formula:

\[
S(w) = \frac{W_{rel} \cdot W_{position}}{W_{case} + \frac{W_{freq}}{W_{rel}} + \frac{W_{sentence}}{W_{rel}}}
\] (2.29)
Then, a contiguous sequence of \( n \)-gram candidate key-phrases is generated using a sliding window. For each candidate key-phrase \( v \) the following score is assigned:

\[
S(v) = \frac{\prod_{w \in v} S(w)}{\text{TF}(v) \ast (1 + \sum_{w \in v} S(w))}
\]

(2.30)

again the lower the score the more meaningful the key-phrase \( v \) is. The authors evaluated the effectiveness of YAKE on 20 datasets against other several state-of-the-art methods such as TF-IDF, KP-Miner [El-Beltagy and Rafea, 2009], TextRank [Mihalcea and Tarau, 2004] and others. In total, YAKE was the best performer on 11 out of 20 datasets, while the second best performing method was KP-Miner on 5 datasets. Similarly to AutoPhrase, YAKE is also language and domain independent making it suitable to a range of tasks where the dataset used is unlabelled. Contrary to AutoPhrase, YAKE does not use a knowledge base nor it is trained on any specific corpora making it suitable for tasks where access to training data is scarce or nonexistent.

### 2.2.6 Evaluation Measures

One of the first and most common methods of evaluating topic modeling is perplexity. However, a study conducted by Chang et al. [2009] showed that perplexity is often negatively correlated with human judgements of topic quality (using topics discovered with LDA and other models), suggesting that evaluation should be focused upon real-world task performance and taking into consideration human annotations. The authors conducted two different experiments. In the first one, named word intrusion, human subjects needed to evaluate if a word is semantically coherent in a given topic (i.e., identifying a spurious word inserted into a topic). The second experiment, called topic intrusion, tests whether the association between a document and a topic makes sense, i.e., the human subject must identify a topic that was not associated with the document. A number of studies have since focused on developing different topic coherence measures, which capture the semantic interpretability of discovered topics, and are aligned with human evaluations of a topic.

This section describes a number of evaluation metrics that have been proposed in the literature to evaluate topic modelling and clustering results, e.g., UCI and NMI, respectively.

**UCI**

For instance, Newman et al. [2010] proposed UCI, an extrinsic measure that uses Pointwise Mutual Information (PMI) to evaluate the coherence of a topic. The term coherence here means that a topic \( T \) (i.e., a list of words \( \langle w_1, \ldots, w_N \rangle \)) should only be composed of related words. For example, a topic \( \langle \text{car}, \text{automobile}, \text{vehicle} \rangle \) is considered coherent, because all terms in this topic refer to the same object or concept, while a topic composed of \( \langle \text{car}, \text{phone}, \text{dog} \rangle \) is not coherent, as these terms do not have any apparent relatedness. This approach is considered an extrinsic metric, because it uses an external dataset to estimate word co-occurrence counts (i.e., these are derived from documents that were not used to build the topics). The counts are constructed by a sliding window (i.e., each sliding windows is considered a document) that moves over the external corpora (e.g., Wikipedia) computing the PMI (see
Equation 2.27 and 2.28) between all pairs of words. Once the PMI scores are computed, the coherence of a can be evaluated, as follows:

$$C_{UCI}(T) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \text{PMI}(w_i, w_j)$$  \hspace{1cm} (2.31)

This metric is bounded between a value of 0 (low coherence) and 1 (high coherence).

**UMass**

On the other hand, Mimno and M. Blei [2011] proposed UMass, arguing that topic coherence evaluation should be based on an intrinsic measure, i.e., the probabilities must be computed based on co-occurrence frequency in the documents used to learn the topics and not on an external corpora as is the case of UCI. Although UMass (Equation 2.33) is similar to the UCI measure, the former introduces an ordering constraint on the topic $T$, i.e., words in the topic are ordered by their importance. In UMass the probabilities are computed differently than UCI, as follows:

$$P(w_i) = \frac{\text{DF}(w_i)}{|D|} \quad P(w_i, w_j) = \frac{\text{DF}(w_i, w_j)}{|D|}$$  \hspace{1cm} (2.32)

In the previous equations, $\text{DF}(w_i)$ is the document frequency where the word $w_i$ occurs (i.e., the number of documents with least one word $w_i$), $\text{DF}(w_i, w_j)$ is co-document frequency of words $w_i, w_j$ (i.e., the number of documents containing one or more words $w_i$ and at least one word $w_j$), and $\epsilon$ is the smoothing parameter that is added to avoid calculating a logarithm of zero. Furthermore, $P(w_i)$ represents the probability of word $w_i$ occurring in a document, while $P(w_i, w_j)$ is the probability of both $w_i$ and $w_j$ co-occurring in a document.

$$C_{UMass}(T) = \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \left( \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \right)$$  \hspace{1cm} (2.33)

Moreover, the UMass coherence is computed based on pairs of consecutive words, instead of considering all possible pairs of words in a topic. This allows one to compute the coherence only considering the top $M$ words, with $M < N$, because the remaining words $(w_{M+1}, \ldots, w_N)$ might not be relevant to the topic. A major drawback of the UMass metric is that it can produce low coherence scores on a small dataset. This is related to the fact that there might not be many occurrences between pairs of words, hence the co-occurrence probability of these pairs will be low. This metrics is upper bounded by value of 0, and values closer to 0 are indicative of high topic coherence.

**CV Measure**

Röder et al. [2015] surveyed the literature of topic coherence measures and found multiple techniques being proposed, although these techniques did not align with the human judgements of a topic. Hence, the authors created a new topic coherence measure labeled CV. After reviewing multiple topic coherence metrics, the authors came to the conclusion that a topic coherence measure should be composed of
four composable steps. Furthermore, the authors suggest that one must follow these steps (framework) when proposing a new topic coherence measure. To better understand the framework, consider again that a topic $T$ is a list of words $\langle w_1, \ldots, w_N \rangle$. In order to calculate the topic coherence of $T$, one should follow the steps described next:

1. **Segmentation**, in which $T$ is segmented into subsets of words $S$, e.g., several subsets of two words as in UCI.

2. **Probability Calculation**, in which the word probabilities $P$ are computed based on a given reference corpus, e.g., through the sliding window approach considered by UCI.

3. **Confirming Measure**, in which both $S$ and $P$ are used to calculate the agreements $\varphi_i$ of subsets of $S$, e.g., using the PMI.

4. **Aggregation**, in which all confirmations $\bar{\varphi} = (\varphi_1, \ldots, \varphi_{|S|})$ of all subset pairs $S$ are aggregated to a single coherence score, e.g., taking the arithmetic or geometric mean of $\bar{\varphi}$.

The authors used CV to measure the correlation within topics on six datasets from distinct domains. They compared the proposed technique against other popular topic coherence measures, such UCI and UMass, and showed that from all CV yielded results which correlate more with human judgements of a topic, than any other coherence measure. Similarly to UCI, this metric is also bounded between a value of 0 (low coherence) and 1 (high coherence). Based on the premise that CV aligns with the human evaluation of a topic, we also make use of CV measure to evaluate the topics generated by the final representation of each model, thus assessing the quality of the generated topics.

**Clustering performance**

One of the most widely used clustering algorithm is the $k$-means, which is an unsupervised clustering method that assigns a set of $n$ data points $(x_1, \ldots, x_n)$ to a pre-defined number of exactly $k$ clusters $\{C_1, \ldots, C_k\}$. Each cluster is defined by its centroid, hence the algorithm starts with initial estimates for the $k$ centroids, which can either be randomly generated or randomly selected from the dataset. The algorithm then iterates between the following two steps:

- **Assignment step**: each data point is assigned to its nearest centroid. Specifically, if $c_i$ is the collection of centroids in set of clusters $C$, then each data point $x$ is assigned to a cluster by:

$$\text{arg min}_{c_i \in C} \text{dist}(c_i, x)^2$$

(2.34)

where $\text{dist}(\cdot)$ is the Euclidean distance.

- **Update step**: in which the centroids are recompute by taking the mean of all data points assigned to that centroid’s cluster as follows:

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

(2.35)

where $S_i$ is the set of data point assignments for each $i$ cluster centroid.
The algorithm iterates between assignment step and update step until a stopping criteria is met, for example, when no data points change clusters, or when the sum of the distances is minimized, or upon reaching a certain number of iterations. Because the k-means algorithm might use a randomized initialization it can converge to a local optimum, thus multiple runs are needed in order to achieve more accurate results.

There are numerous ways to evaluate the results produced by k-means algorithm, for instance, by comparing the information shared between two clusterings, through the Normalized Mutual Information (NMI) metric. Let us start by assuming that each data point has an equal probability of being picked, then the probability of the outcome being in cluster $C_k$ equals to:

$$P(k) = \frac{n_k}{n}$$  \hspace{1cm} (2.36)

where $n_k$ refers to the total number of points assigned to cluster $C_k$ and $n$ is the total number of data points, and let $P(k, k')$ represent the probability of a data point belonging to a cluster $C_k$ in clustering $C$ and to cluster $C_{k'}$ in clustering $C'$, i.e., the joint distribution of random variables associated with the two clusterings:

$$P(k, k') = \frac{|C_k \cap C_{k'}|}{n}$$  \hspace{1cm} (2.37)

The NMI measures the information shared by two clusterings $C$ and $C'$, defined as follows:

$$NMI(C, C') = \frac{I(C, C')}{H(C) + H(C')/2}$$  \hspace{1cm} (2.38)

where,

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \left( \frac{P(x, y)}{P(x) P(y)} \right)$$  \hspace{1cm} (2.39)

is the mutual information between the variables $X$ and $Y$, while

$$H(X) = -\sum_{x \in X} P(x) \log(x)$$  \hspace{1cm} (2.40)

is the information entropy for a discrete variable $X$. More specifically, NMI estimates the quality of clustering results $C$ against the ground truth $C'$, varying between 0 when there is no information shared between the resulting cluster and the ground truth to a value of 1 when there is perfect correlation between ground truth and clustering result achieved by a clustering algorithm such as k-means.

### 2.3 Related Work

With the advancements in EHR system, large quantities of data becomes available to analyze and extract information contained on it, in order to aid healthcare professionals take decisions faster and better. This section gives an literature overview of the architectures proposed in the recent years, that aim at encoding the medical data and extracting useful relationships by creating latent representation
of textual data from the medical domain. These architectures were capable of uncovering relationships between complications which were confirmed by medical literature findings [Bhattacharya et al., 2016, Gefen et al., 2018], thus proved to be of great significance for the healthcare domain, where the amount of newly created information can not be solely processed by a healthcare professional. Hence, the recent developments in Deep Learning (DL) and Natural Language Processing (NLP) fields allows to create architectures are capable of aiding a healthcare professionals to process these large quantities of data.

2.3.1 Building Representations for Biomedical Text

One of the main issues facing pre-processing of data from EHR system is high-dimensionality and high-variability (includes a wide range of data such as diagnosis, medication, test results), temporality which refers to the sequential nature of clinical events and limited interpretability due to the complexity of medical data. To tackle one or several of these issues, multiple teams proposed methods based on topic modelling techniques, such as LDA [Bhattacharya et al., 2016] and LSA [Gefen et al., 2018], while other teams proposed deep learning methods based on autoencoders and extensions [Gencoglu, 2019, Sadati et al., 2019, Way and Greene, 2017, Zhang et al., 2019], these are known as representational learning and are used to identify the relevant features present in the data during the pre-processing stage, removing the noise and enhancing the downstream task. Another widely used approach is word embeddings, which has been used to extract medical terms [Vine et al., 2015] or to encode clinical events (such as health center visits or clinical procedures) allowing to compute similarities between patients [Choi et al., 2016, Zhu et al., 2016].

Based on Topic Modelling

Work conducted by Bhattacharya et al. [2016] focused on identifying co-occurrences of health conditions in EHRs using topic modelling approaches. They showed that LDA is capable of capturing hidden patterns, e.g., between diabetes and function decrease of kidneys, among other interesting associations. The topics retrieved were validated by medical literature and proved to relate to real world co-occurrence of health conditions. The dataset used by the authors contained structured data with attributes such as age, ethnicity, gender, lab test results, and diagnosed conditions (represented as Systematized Nomenclature of Medicine (SNOMED[^3]) codes) of patients that have shown evidence of decrease in kidney function. The authors used the codes associated to diagnosed conditions as input to the LDA model, hence each topic \( T = \{c_1, c_2, \ldots, c_N\} \) is a distribution over SNOMED codes. The authors proposed to evaluate the quality of the resulting topics, based on two metrics:

- **Tightness**: measures if a topic can be represented by a small number of codes, e.g., by showing that a topic can be defined by ten or fewer codes (i.e., the weights of the top ten codes account for more than 95% of the probability in a topic).

[^3]: [https://www.snomed.org/](https://www.snomed.org/)
• **Distinctiveness**: is measured by computing the inter-topic distance, e.g., by assessing if pairs of topics are separated by a high Jensen-Shannon (JS) divergence.

Let us consider that the KL divergence is formulated as follows:

$$D_{KL}(P \parallel Q) = \sum_{x=1}^{N} P(x) \log \left( \frac{P(x)}{Q(x)} \right)$$  \hspace{1cm} (2.41)

then the distinctiveness between the topics $P$ and $Q$ (i.e., probability distributions over SNOMED codes) can be computed using the JS divergence, which is a smoothed version of the KL divergence, computed as follows:

$$D_{JS}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M)$$  \hspace{1cm} (2.42)

In the previous equation, $M$ is the mean probability distribution $M = \frac{1}{2} (P + Q)$ between $P$ and $Q$. The range of values for $D_{JS}$ varies between 0 and $\log(2)$, where 0 indicates identical distributions, and $\log(2)$ means non-overlapping topics. The authors argue that tightness and distinctiveness metrics enable one to measure the relevance of the topics found, identifying relationships between codes in a topic or inter-topics.

Gefen et al. [2018] further confirmed claims stated by Bhattacharya et al. [2016], showing that topic models can be used to identify relationships between health conditions. The dataset used by these authors was composed of 32,124 medical records from 416 distinct patients collected between 2013 and 2014. The authors built a VSM, i.e., a term-document matrix representation composed of medical descriptions and ICD-10 Clinical Modification (ICD-10-CM) codes retrieved from the medical records, where each cell in the matrix contains the TF-IDF weight that scores the association between a term and a document. This matrix was reduced by applying LSA, resulting in a dense topic matrix containing medical terms and codes. The cosine distance (Equation 2.2) was used to measure the degree of similarity between medical terms and ICD-10-CM codes. The considered approach was capable of identifying a strong relationship between *hypertension* and *gastroesophageal reflux disease*, and this relationship was only recently confirmed by findings in medical literature.

Our work is closely related to the works of Bhattacharya et al. [2016] and Gefen et al. [2018], as we also employ a topic quality measure (CV) similar to that of Bhattacharya et al. [2016], although instead of using JS divergence (usually used for probability distributions) the CV coherence uses the cosine similarity which is more adequate to our methods. Because the methods employed by us create vectors and not probability distributions as is the case of LDA.

**Based on Word Embeddings**

Word embeddings have been proved to be powerful approach to capture implicit similarity between words, as described on Section 2.2.3. For this reason, multiple teams proposed architectures that leverage word embeddings during the data representation stage, enhancing and condensing the initial input before applying a downstream task. For instance, Vine et al. [2015] have shown that by employing word embeddings improves the information extraction effectiveness. They also demonstrate that the
corpora used to generate the embeddings takes an important role on the effectiveness, i.e., training on a large healthcare corpora yields better results than large generalist one. Hence, one must take cautious measures when employing word embeddings as representation layer, as the results might be affected by the data in which the word embedding training took place.

One a similar line of work, Choi et al. [2016] presented a new multi-layer architecture, named Med2Vec, that learns both code-level and visit-level representations leveraging word embeddings. Med2Vec is capable of learning code-level representations contained in EHRs by employing the skip-gram model (see Section 2.2.3) based on co-occurrences of codes. This yields the final code representation as a non-negative matrix $W_c$. The authors argue that this representation which is similar to the results of the NMF, and is highly interpretable, which is crucial in healthcare applications, where understanding the obtained results is very important, and the performance of the model can be compromised. The authors consider a visit to be comprised of multiple medical codes, and hence the visit-level representation is learned by concatenating demographic information with a binary vector (encoding the occurrence of all medical codes in the visit). Figure 2.6 depicts the new architecture proposed by these authors.

As mentioned previously, this new architecture aims to enhance the interpretability of results by keeping the final code representation non-negative. To achieve this, Med2Vec makes use of a rectifier linear unit (ReLU) function instead of a liner function, defined as follows:

$$\text{ReLU}(v) = \max(v, 0)$$ (2.43)

where the $\max$ function is applied element-wise to the vector $v$, after which it only contains positive values including zero, i.e., all negative entries in the vector are transformed into zero. The Med2Vec architecture (depicted in Figure 2.6) uses the ReLU function to transform a visit $x_t$ to a visit representation $v_t$, as a two step process:
1. The input data $x_t$ is transformed by a function, ReLU($\cdot$), to an intermediate visit representation $u_t$:

$$u_t = \text{ReLU}(W_c x_t + b_c)$$  \hspace{1cm} (2.44)

2. Then, the demographic information (e.g., the age of a patient), represented by a vector $d_t$, is incorporated into Med2Vec by concatenating it with $u_t$. A second ReLU($\cdot$) function is applied, resulting in the final visit representation $v_t$:

$$v_t = \text{ReLU}(W_v [u_t, d_t] + b_v)$$  \hspace{1cm} (2.45)

After this process, Med2Vec uses $v_t$ to train a multi-layer perceptron that predicts neighboring visits $(\ldots, x_{t-2}, x_{t-1}, x_{t+1}, x_{t+2}, \ldots)$ normalizing the probability distribution of the predicted visits using a softmax function (Equation 2.15). The authors used two distinct datasets to evaluate the architecture from two perspectives, namely the quality of code-level representations were analyzed by medical experts and through NMI (see Section 2.2.6) and second perspective was a classification task to evaluate visit-level representations by predicting the medical codes assigned in the next visit. The authors compared Med2Vec against other representation learning methods, such as SVD and skip-gram, and shown that their architecture produces more interpretable code-level and visit-level representations while remaining stable to hyperparametrization, contrary to skip-gram model which produced different when distinct initializations where in place.

Recently and in a similar line of work, Zhang et al. [2019] developed a novel method to encode and represent medical terms given an external knowledge. Although this method, named Conceptual Contextual (CC) Embeddings, is more related to BERT architecture, proposed by Devlin et al. [2019], than Word2Vec. Furthermore, CC embeddings is adapted to the medical domain by allowing to use Unified Medical Language System (UMLS) as external source to enhance the representation of medical concepts. After CC embedding were trained these were used in two downstream classification tasks, specifically, readmission and mortality prediction and achieving state-of-the-art results, outperforming previous methods in the medical domain.

Based on Neural Networks

More recently, Sadati et al. [2019] compared multiple deep learning architectures, based on autoencoders variants, on three datasets from the medical domain, namely adversarial autoencoder, stacked sparse autoencoder (SSAE) and Variational Autoencoder (VAE) and deep belief networks, showing how each architecture affects the final results in a supervised learning setting. More specifically, the authors compared the performance of each one on a classifications task. The best performing architecture was SSAE on a small dataset and while VAE performed the best on two large datasets. The authors argue that the feature selection architecture should be strongly influenced by the size of the dataset that is used in order to achieve the best results.

In a similar line of work, Gencoglu [2019] proposed a deep neural network architecture for learn-
ing compact representations of health events. The proposed architecture leverages word embedding representations, these then are fed to a deep convolutional autoencoder (CAE) network. The authors compared their technique against conventional representation methods, including bag-of-words with frequency weights. The architecture implementation allows to be flexible enough to accept different word embeddings techniques (e.g., Word2Vec or GLoVe). The authors evaluated their CAE-based network against LDA and NMF, using a dataset composed of tweets from multiple health-related sources, e.g., from BBC Health. The proposed architecture produced impressive results in clustering tasks, when considering the nature of a tweet, i.e., a brief description and without a label associated.

2.3.2 Healthcare Applications

In the recent years with the introduction of newer and better performing autoencoder variants, multiple teams have applied these architectures to the bioinformatics domain, as these architectures are suitable for dimensionality reduction of high dimensional data, as is the case of genomic data. For instance, Way and Greene [2017] compared VAE against evaluated NMF and PCA on gene expression of pan-cancer data. More specifically, the authors evaluated the performance of VAE to generate and to predict gene inactivation, showing that this method is much better at capturing the signals that lead to gene inactivation.

Other teams have addressed the tasks of automatically assigning ICD-9 [Miranda, 2018] and ICD-10 [Duarte et al., 2018] codes to clinical text (e.g., hospital discharged summaries or death certificates), reducing the time involved in coding and helping healthcare professionals with assignment hints. Duarte et al. [2018] built a complete and complex neural network architecture composed of averaging word embeddings, and a two-level hierarchical arrangement of neural attention mechanisms and gated recurrent units (GRUs), to classify free-text electronic medical records. Duarte et al. [2018] reported impressive results at classifying death certificates written in Portuguese language at the ICD-10 code-, block- and chapter-levels. Miranda [2018] focused on further refining the architecture, proposed by Duarte et al. [2018], to classify free-text hospital discharge summaries in Portuguese language. The task was also a multi-label classification problem, but considering ICD-9 codes. Furthermore, the architecture also needed to adapt and deal with additional structured data that contains relevant information about a patient (e.g., age group of the patient or the department of stay) which allows to create more robust model.

2.4 Overview

This chapter presented the theoretical foundations of several textual representation methods (Section 2.2.1), dimensionality reduction techniques (Section 2.2.2), key-phrase extraction techniques (Section 2.2.5) and feature and representation learning approaches (Section 2.2.3 and Section 2.2.4) which we employ in our experiments and their respective evaluation metrics (Section 2.2.6) to estimate the quality of the results produced. Furthermore, this chapter described the state-of-the-art methods
proposed in the literature to leverage the extraction and representation of medical terms or medical events, and from which this work takes inspiration. This chapter also briefly introduced a small number of healthcare applications proposed by multiple teams, powered by deep learning architectures, that are aimed automatically classifying causes of death and hospital discharge summaries, thus alleviating the burden of healthcare professionals in this task.
Chapter 3

Proposed Approach

This work proposes an architecture that given a collection of documents in free-text allows to extract and represent the main features contained in them, where features are the most relevant classification codes and phrases that were identified by the architecture. The following sections detail the steps taken to achieve this. Starting with an introduction of the text pre-processing and key-extraction pipeline, followed by description of multiple techniques employed to transform the original sparse text space into a vector space representation, which can be further processed by dimensionality reduction and representation learning techniques to firstly condense the data and then extract the relevant features contained on the data.

3.1 Building Representations for Documents and Key-Phrases

As it is common in text processing, the first stage is the pre-processing step aiming to remove the noise from data as well as normalizing it, and in our case it consisted on stop word removal and lowercase transformation. The stop words corpus contained 573 tokens, composed of commonly used words in the Portuguese language, as well periods of the day (i.e., manhã, tarde and noite), and a subset of frequent numbers in both algebraic and grammatical (i.e., 5 and cinco). These new tokens added to the stop word corpus were necessary, because these are frequently used by healthcare professionals to mark patient’s admission by specifying the time or periods of the day. After stop word removal, all documents are fed to the AutoPhrase framework in order to uncover the most relevant key-phrases for each document. We used a subset of Wikipedia database dump in Portuguese\(^1\), only containing the pages’ title as knowledge base for AutoPhrase. Meanwhile, for the quality phrase re-estimation stage, we provided the ICD-10 Clinical Modification (ICD-10-CM) code descriptions in Portuguese language, which contains terms from the medical domain and more related to the dataset used in this work, thus assigning higher weights to medical terms. The Portuguese version of ICD-10-CM that we used is publicly available from Centro Terminologias Clínicas\(^2\) website.

However, before proving the Wikipedia pages’ title as knowledge base, we applied a pre-processing

\(^1\)https://dumps.wikimedia.org/ptwiki/latest/
\(^2\)https://interop-pt.atlassian.net/wiki/spaces/CTCPT/pages/58884216/Projetos
step to remove titles composed of only numbers, dates, Internet Protocol (IP) addresses, emails or unique page identifiers which are used to identify a discussion page. Although initially there were 4,913,002 titles, after pre-processing resulted in a total of 3,662,864 titles containing 9,904,455 words.

The pre-processing step of the knowledge base provided for the quality re-estimation stage (i.e., ICD-10-CM code descriptions in Portuguese) was much simpler, we only applied lowercase transformation in order to have consistent casing between the original data and the quality phrases. In this knowledge base there were a total of 93,830 descriptions containing 1,016,625 words.

After the AutoPhrase stage, the original documents and key-phrase extracted are transformed to a vector space representation (step 3 in Figure 3.1), leveraged by either TF-IDF terms score or document embeddings based on Doc2Vec. In order to compare both vector space representation experimented, the pre-processing and key-extraction stages of the documents are equal, thus minimizing any variance that could be introduced in these stages.

Diving more deeply into specific details, to create the TF-IDF representation we considered a $n$-gram range from 1 to 3, this was necessary since a majority of clinical terms are composed of several words/tokens (e.g., acidente cardiovascular or insuficiência renal crônica) hence by picking this range allows to capture all words that compose a clinical term.

To build the Doc2Vec representation we used the Gensim library developed by Řehůřek and Sojka [2010]. We followed the recommendation of the original authors of Doc2Vec and used a combination of PD-DM and PD-DBOW, considering the context window size of 3 words, while the size of paragraph (document) vector was set to 2000 and trained for 500 epochs with a learning rate of $\eta = 0.001$.

Once the intermediate vector representations explained previously are obtained, these are further processed by dimensionality reduction methods. The methods experimented can be divided into two classes, these can be based on the NMF and deep neural networks, as is the case of DSNMF and AE. All models based on NMF were implemented using the scikit-learn\(^3\) library, meanwhile all neural network models where implemented using the Tensorflow\(^4\) library. The SNMF method was also a Tensorflow implementation based on the version used by Trigeorgis et al. [2014].

### 3.1.1 Methods based on NMF

In this work, we propose to compare the results achieved by simple factorization approaches (NMF and SNMF) against deep neural networks in representation learning task, specifically considering a topic modelling and a clustering settings. Based on the premise that neural network approaches are capable of non-linear transformation of the input data producing better results in these settings, we propose to run experiments with two distinct deep feedforward neural network based dimensionality reduction architectures, namely DSNMF and AE.

\(^3\)https://scikit-learn.org/stable/index.html
\(^4\)https://www.tensorflow.org
Taking inspiration from representation learning techniques presented in Section 2.3, we present our proposed approach which is based on matrix factorization techniques (as is the case of NMF and SNMF) and on deep neural networks (as is the case of AE and DSNMF) described in Section 2.2.2 and Section 2.2.4, respectively. Our approach is composed of three stages, and a visual description of the architecture is depicted in Figure 3.1. The first two stages of the architecture were already explained in the previous section (see Section 3.1).

In the third stage, we apply every method over the intermediary representation (step 4 in Figure 3.1), which yields the final latent representation (step 5). This final representation is then evaluated based on similarity between the features extracted, and then more indirectly from a topic modelling and a clustering perspective. Follows a introduction to the methods and parameters used during our experiments and their respective implementation details and sources.

In our study, we used the open-source implementation of NMF available from the scikit-learn\(^5\) package, which implements the fast multiplicative update solver, proposed by Févotte and Idier [2011]. We also used the a dense variant of the Nonnegative Double Singular Value Decomposition initialization method (NNDSVDa) for matrices \(W\) and \(H\), based on two SVD processes in which one approximates the data matrix, and the other approximates positive sections of the resulting partial SVD factors, utilizing an algebraic property of unit rank matrices [Boutsidis and Gallopoulos, 2008]. In our experiments, we used both the Frobenius norm and Kullback-Leibler divergence, which we refer to as FR-NMF and as KL-NMF, respectively. These methods were trained for a maximum of 200 epochs, or stopping early with a tolerance \(\theta = 0.005\) between epochs when there was no improvement in the reconstruction error.

The NMF based methods are also parametrized by a $\alpha$ and $l_1$ ratio which we used the default values of 0 for both.

All the techniques used in our experiments were tested considering multiple values for the final representation parametrized by $\tau \in \{5, 10, 15, 20, 25, 30, 40, 50, 100, 150, 200\}$, this allows us to verify how the different dimensionalities affect the final performance, hence assessing if there are particular techniques which perform better when considering higher or lower dimension for the latent representation.

### 3.1.2 Methods based on Neural Networks

Although initially both DSNMF and Autoencoder (AE) techniques (Section 2.2.4) were crafted for image processing pipelines, we set to explore how these techniques can be extended and applied in text processing. Specifically, the DSNMF produced impressive results on clustering of human faces, being capable of learning hierarchical features, such as face pose or expressions, at different layers of the neural network. Meanwhile, the authors of AE [Hinton and Salakhutdinov, 2006] have applied this technique to text processing, showing better results than LSA in document retrieval tasks, demonstrating once again that non-linear transformation yielded better results than linear ones.

In the experiments, we used the same parameters for both DSNMF and AE, both having a two hidden-layer network, with 2000 as the size of the first hidden layer while the second layer (i.e., the latent representation) was also parametrized by $\tau$, we used ReLU($\cdot$) as activation function, and the neural network was trained through the Adam optimizer [Kingma and Ba, 2014] with a learning rate of $\eta = 0.001$ for a 100 epochs and with a cost tolerance of $\theta = 0.05$ between epochs as stopping condition. Both methods are constituted of deep feedforward neural networks, having the weights of the network initialized using the Xavier initialization technique, proposed by Glorot and Bengio [2010], which showed that using a adequate initialization technique reduces the need to pre-train the network. Similar to the NMF based methods, here we also tested different values for the latent representation $\tau$ in order to compare all methods across different latent dimensionalities, doing so, allows us to verify if these methods display different behaviors when other dimensions are used.

In our experiments, the SNMF and DSNMF used were adapted versions of the open-source implementation\(^6\) using the Tensorflow library, based on original proposal by Trigeorgis et al. [2014]. Similarly, we used an adapted version of the open-source AE implementation available from the Tensorflow research team\(^7\). The adaptations were necessary in order to standardize the Application Programming Interface (API) provided by the implementations that use the Tensorflow library to the ones using the scikit-learn library, i.e., to be compatible with pre-existing data flow in architecture pipeline.

### 3.2 Overview

This chapter detailed the techniques considered into the architecture that proposed in this work, which enables us to process, extract and represent textual information. Furthermore, it also describes the

---

\(^6\)https://github.com/yvettegwy/deep-semi-nmf/

\(^7\)https://github.com/tensorflow/models/tree/master/research/autoencoder
specific implementations and parameters considered for every method experimented.
This chapter commences by introducing the dataset and its exploratory analysis that took place before using it as input to the architecture in Section 4.1.1, describing the steps taken to convert the original documents encoded with ICD-10 classification system to ICPC-2 in Section 4.1.3, proceeding to explain the methodology employed to evaluate the different methods employed in Section 4.1.4 and the discussing results produced by these methods in Section 4.2.

4.1 Datasets and Evaluation Methodology

This section describes the metrics used to evaluate the architecture on a dataset consisting of free-text clinical descriptions in Portuguese. But first, this section gives the statistical characterisation of the dataset obtained after data analysis and exploration, then explains the process that consisted on augmenting (transforming) the original dataset with ICPC-2 classification codes and descriptions, and finally introducing the evaluation methodology employed to measure all methods tested during our experimentation.

4.1.1 Dataset

The dataset used is from the medical domain, and is composed of 206 644 death certificates, henceforth referred to as documents, collected between the years of 2013 and 2016. Each document contains a brief free-text description for the cause of death in Portuguese alongside the respective ICD-10 code for the main cause of death, as well as ICD-10 codes for auxiliary/contributing conditions (see Table 4.1). Both the free-text description and ICD-10 codes were written and labelled by a healthcare professional. The dataset used does not have any fields containing personal identifiable information, hence no pre-processing was needed to anonymize the data.

Furthermore, originally the dataset was not labelled according to the ICPC-2 terminology, thus a transformation of the dataset was in place in order to create the mapping between the ICD-10 and the ICPC-2 systems, as detailed in Section 4.1.3. In Table 4.1 are present five documents displaying the structure of the dataset used in this work. Although originally it did not have the ICD-10 Chapter and
ICD-10 Main | ICD-10 Auxiliary | Description | ICD-10 Chapter | ICD-10 Block
---|---|---|---|---
E86 | N179 G61 N390 | uresepsis insuficiencia renal aguda desidrata... | 4 | 9
R54 | I248 | sindrome coronario agudo- 2 5 senilidade- 2 4 | 18 | 8
E86 | N179 I694 J960 | insuficiencia renal aguda desidratacião insufic... | 4 | 9
R54 | I350 | senilidade estenose arortica insuficiencia car... | 18 | 8
R688 | G310 | falência de funções atrofia cerebral | 18 | 8

Table 4.1: Dataset structure before transformation.

<table>
<thead>
<tr>
<th>Number of documents</th>
<th>206 444</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average words per document</td>
<td>12.1</td>
</tr>
<tr>
<td>Vocabulary size</td>
<td>56 823</td>
</tr>
<tr>
<td>Number of distinct ICD-10 main codes</td>
<td>2 285</td>
</tr>
<tr>
<td>Number of distinct ICD-10 codes</td>
<td>4 485</td>
</tr>
<tr>
<td>Number of distinct ICD-10 chapters</td>
<td>20</td>
</tr>
<tr>
<td>Number of distinct ICD-10 blocks</td>
<td>212</td>
</tr>
<tr>
<td>Number of documents mapped to ICPC-2</td>
<td>8 452</td>
</tr>
<tr>
<td>Number of distinct ICPC-2 codes</td>
<td>65</td>
</tr>
<tr>
<td>Number of distinct ICPC-2 chapters</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 4.2: Statistical characterization of the dataset.

ICD-10 Block columns, these where calculated later based on the ICD-10 Main column. The ICD-10 Chapter and ICD-10 Block fields are only used for data analysis, exploration and visualization of the results, thus are not feed as attributes to the architecture. The ICD-10 classification system is organized in chapters, where each chapter contains several subsets which are called blocks. Hence, when referring to block one must also provide the respective chapter. Meanwhile, the ICPC-2 classification system is only subdivided into chapters.

During the initial exploration of the dataset, we noticed that average length (number of terms) of the free-text descriptions is 12.1 with a standard deviation of 26.30, where 75% of documents have less than 12 terms. This suggests that each description is very brief, containing relatively low number of words. For example, one can notice the Description column in Table 4.1, which shows that each document has a condensed number of medical terms. Taking into account all the documents the vocabulary of the dataset contains 56 823 unique words. Table 4.2 presents more statistical information regarding the dataset, including information about the ICPC-2 classification system which were only obtained after the transformation stage, as explained in Section 4.1.3.

### 4.1.2 Dataset Exploration

The dataset used in this work was also used by Duarte et al. [2018] in a multi-label classification setting, as detailed in Section 2.3.2. Duarte et al. [2018] found evidence of many problems in terms of spelling and non-standard abbreviations, and thus terms like *insuficência* and *insuficiência* are considered different terms, although the former is a misspelled version of the latter. Regarding the distribution of ICD-10 codes, shown in Figure 4.1, the top most frequent codes are related to the respiratory system (J189 and C349), cardiovascular diseases (I64 and I509), and unspecified causes of death (R99).

One can notice that the dataset is highly imbalanced with the top 20 codes accounting for over 30%
of documents in the dataset. To better understand the distribution of the documents, let us also visualize how the documents are distributed across ICD-10 Chapters and Blocks, as displayed in Figure 4.2. Although the heatmap visualization shows 29 blocks for every ICD-10 chapter, in reality the majority of chapters have less than 15 blocks, however for visualization purposes we set to zero the invalid block entries. From the heatmap, one can notice similar imbalance, where the top 5 predominant (by order of document frequency) chapters are Chapter 9 (circulatory system complications), Chapter 2 (neoplasms), Chapter 10 (respiratory system complications), Chapter 18 (symptoms, signs, abnormal results of clinical or other investigative procedures, and ill-defined conditions) and Chapter 1 (infectious and parasitic diseases), in this order.

In order to get more insight on the actual content of the dataset, let us visualize which are the most frequent terms contained in the dataset using a word cloud as it is displayed in Figure 4.3. The majority of these terms are related to the cardiovascular, respiratory and urinary systems, describing complications such as hipertensão arterial, doença pulmonar and renal crónica, respectively. This observation is supported by the ICD-10 code and chapters distribution previously mentioned.

### 4.1.3 Dataset Transformation

In this work we proposed to compare the hierarchical structure of two medical classification systems, namely ICD-10 and ICPC-2. However, originally the dataset was only labeled with codes from the former, thus in order to compare the two, the dataset needed to be transformed to include the respective ICPC-2 code. Luckily, based on the ICD-10 codes we are able to directly map them to a ICPC-2 codes. However, there are no resources available to achieve this mapping directly in Portuguese language, thus
Figure 4.2: ICD-10 chapter and block distribution. Showing the high imbalance in the dataset, as the chapters 5, 8, 15 and 16 combined account for only 64 documents, while all blocks from chapter 2 account for 44,890 documents.

Figure 4.3: Word cloud of the most common terms based on their frequency in the dataset.

to achieve this mapping we employed a two step process.

In the first step, we used the English language version of ICPC-2 (i.e., ICPC-2E version 7) available at [ehelse.no](https://ehelse.no/kodeverk/icpc-2e--english-version)¹, to build a translation table that maps every ICPC-2 code to its ICD-10 counterpart. In the second step, this intermediary translation table was enhanced with the respective ICPC-2 description in Portuguese also available at [ehelse.no](https://ehelse.no/kodeverk/icpc-2e--english-version#ICPC-2e%20translations)², then through this intermediary translation, every original document was enhanced with the respective ICPC-2 code and description in Portuguese.

The mapping in the translation table is many-to-many, as shown in Table 4.3, i.e., one ICPC-2 code can be mapped to several ICD-10 codes and vice-versa. For instance, A03 maps to both codes R508 and R509, while A04 and A05 map to R53. After these transformations the structure of the dataset is

¹https://ehelse.no/kodeverk/icpc-2e--english-version
²https://ehelse.no/kodeverk/icpc-2e--english-version#ICPC-2e%20translations
displayed in Table 4.4, which shows that every document mapped now has two new columns, namely ICPC-2 Code and ICPC-2 Description.

<table>
<thead>
<tr>
<th>ICPC-2 Code</th>
<th>ICPC-2 Description</th>
<th>ICD-10 Code</th>
<th>ICD-10 Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A03</td>
<td>Fever</td>
<td>R508</td>
<td>Other specified fever</td>
</tr>
<tr>
<td>A03</td>
<td>Fever</td>
<td>R509</td>
<td>Fever, unspecified</td>
</tr>
<tr>
<td>A04</td>
<td>Weakness/tiredness general</td>
<td>R53</td>
<td>Malaise and fatigue</td>
</tr>
<tr>
<td>A05</td>
<td>Feeling ill</td>
<td>R53</td>
<td>Malaise and fatigue</td>
</tr>
</tbody>
</table>

Table 4.3: Mapping snippet between the classification systems ICPC-2 and ICD-10 which only contain description in English language.

Moreover, the ICPC-2 does not cover all ICD-10 codes, only mapping certain ICD-10 chapters and blocks, for instance the ICPC-2 chapters H (ear related conditions), W (family planning) and Z (social problems) where left un-mapped because given the nature of our dataset there were no documents containing such ICD-10 chapters or blocks (see Figure 4.5). It was possible to map a total of 8 452 documents containing 65 distinct ICPC-2 codes. The distribution of the top 20 most common is shown in Figure 4.4, one can notice that the dataset is highly imbalanced (similarly to ICD-10 distribution in Figure 4.1) having P05 (senility) accounting for 35% of occurrences, followed by A02 (chills), A08 (swelling), B29 (lymph) and B04 (blood symptom/complaint).

One must notice that we did not use the columns ICPC-2 Code or ICPC-2 Description as separate input attributes to the architecture, these columns were only used to create the code-pairs and phrase-pairs collections, which allowed us to compare the intra-chapter and inter-chapter similarities of both classification systems, as detailed in the next section.

During our experimentation, we used all documents mapped to ICPC-2 and the remaining 197 992 documents were filtered to only include ones that were labelled with a ICD-10 code with a frequency higher than 10, from these we randomly sampled 50 000. After removing duplicate documents if the ICD-10 code assigned, free-text description and key-phrases extracted were repeated, yielded a total of 41 219 documents, henceforth referred to as $DS_1$. We only sampled 50 000, because it was a hardware limitation when running the DSNMF method, as the implementation of this method is computationally heavy during the transformation of the input, and can not be done in several batches as is the case of AE. For testing purposes and to verify if increasing the number of documents would improve the results, we experimented the other methods (FR-NMF, KL-NMF, SNMF and AE) with all documents (206 644) but there was not a significant improvement in the results, these methods achieved comparable results when considering the sample $DS_1$ consisting of only 41 219 documents.

<table>
<thead>
<tr>
<th>ICD-10 Main</th>
<th>ICD-Auxiliary</th>
<th>Description</th>
<th>ICD-10 Chapter</th>
<th>ICD-10 Block</th>
<th>ICPC-2 Code</th>
<th>ICPC-2 Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E86</td>
<td>N179...</td>
<td>urosepsis...</td>
<td>4</td>
<td>9</td>
<td>T11</td>
<td>Distúrbios metabólicos</td>
</tr>
<tr>
<td>R54</td>
<td>I248</td>
<td>sindrome...</td>
<td>18</td>
<td>8</td>
<td>P05</td>
<td>Sintomas e sinais gerais</td>
</tr>
<tr>
<td>E86</td>
<td>N179...</td>
<td>insuficiencia...</td>
<td>4</td>
<td>9</td>
<td>T11</td>
<td>Distúrbios metabólicos</td>
</tr>
<tr>
<td>R54</td>
<td>I350</td>
<td>senilidade...</td>
<td>18</td>
<td>8</td>
<td>P05</td>
<td>Sintomas e sinais gerais</td>
</tr>
<tr>
<td>R688</td>
<td>G310</td>
<td>falência...</td>
<td>18</td>
<td>8</td>
<td>A02</td>
<td>Sintomas e sinais gerais</td>
</tr>
</tbody>
</table>

Table 4.4: Dataset structure after ICPC-2 transformation, showing the new columns ICPC-2 Code and ICPC-2 Description mapped from the ICD-10 Main column.
Figure 4.4: ICPC-2 code distribution in the dataset of Portuguese death certificates. Portraying the P05 ICPC-2 code associated with senility being mapped in over 3000 documents.

Figure 4.5: ICPC-2 per chapter distribution across all documents mapped.
4.1.4 Evaluation Methodology

We decided to evaluate all methods from three different perspectives. One perspective is to verify if the clinical terms and codes used by healthcare professionals align with the hierarchical structure of ICD-10 and ICPC-2 classification systems, this can be achieved by measuring the cosine similarity between collection of pairs of clinical terms (phrases) and codes. To create these collections, each pair must be composed of two distinct codes, furthermore each pair of codes has to be strictly from either classification system, i.e., a pair with one code from ICD-10 and the other ICPC-2 should not occur, neither be compared. To compare the phrases extracted, we used their respective ICD-10 or ICPC-2 codes to build the phrase-pairs. Enforcing these constraints, the following 12 collections of pairs were created, these are:

- Pairs of ICD-10 codes within the same chapter (ICD SC)
- Pairs of ICD-10 codes within the same chapter and block (ICD SB)
- Pairs of ICD-10 codes from different chapters (ICD DC)
- Pairs of ICD-10 codes from different blocks (ICD DB)
- Pairs of ICPC-2 codes within the same chapter (ICPC SC)
- Pairs of ICPC-2 codes from different chapters (ICPC DC)
- Pairs of phrases with ICD-10 code assigned within the same chapter (P ICD SC)
- Pairs of phrases with ICD-10 code assigned within the same chapter and block (P ICD SB)
- Pairs of phrases with ICD-10 code assigned from different chapters (P ICD DC)
- Pairs of phrases with ICD-10 code assigned from different blocks (P ICD DB)
- Pairs of phrases with ICPC-2 code assigned within the same chapter (P ICPC SC)
- Pairs of phrases with ICPC-2 code assigned from different chapters (P ICPC DC)

After creating these collections, we sampled 330 random pairs from each collection and computed the similarity between each pair by using their respective latent representation. We have only considered 330 pairs because it was the maximum number of codes mapped in the P ICPC SC collection, as there were only 65 distinct ICPC-2 codes and many of them from the different chapters. The average cosine similarity results achieved by each method for code-pairs and phrase-pairs collections are presented in Table 4.5 and Table 4.6, respectively.

The second perspective is related to topic modelling, i.e., to measure how consistent are the resulting topics created by each methods experimented. This can be achieved by calculating topic coherence scores such as CV measure proposed by Röder et al. [2015], which capture the semantic interpretability of the discovered topics and that align with human judgement of a topic, as explained in Section 2.2.6. The coherence of a topic is based on the topic’s top 10 words and shows how strongly pairs of these
words support each other within the given external corpus. In our case we used the documents that were not use in the training as external corpus to compute the co-occurrence counts for the CV coherence score considering a sliding window of 110, as recommended by authors of CV measure. The implementation of the CV coherence measure is offered by the Gensim library [Rehůřek and Sojka, 2010].

The third perspective is to measure the resulting latent representation in terms of clustering performance. We compare all methods on both on supervised metrics, namely Normalized Mutual Information (NMI) and Homogeneity, and on unsupervised metrics, specifically Davies-Bouldin Index (DBI) and Silhouette coefficient. The supervised metrics allows us to evaluate if there is any correlation between ground truth labels and labels assigned by the clustering algorithm (NMI) and if there are clusters that only contain data points of a single class (Homogeneity). While the unsupervised metrics measure the structure of the clustering, for instance the ratio between intra-cluster and inter-cluster distances with DBI (a value of 0 is best) and measure if there are overlapping clusters with Silhouette coefficient ranging from worse at -1 to best at 1, while a value of 0 is indicative of overlapping clusters, negative values indicate that samples were assigned to the wrong cluster, as a different cluster is more similar.

4.2 Experimental Results

This section presents a comparison of the quality of representation obtained by experimental methods described previously. The quality can be assessed by answering these question: i) Which technique best preserves the hierarchical structure of ICD-10 and ICPC-2? and ii) Which technique performs the best from a topic modelling and clustering perspective?. The first question is answered by measuring the similarity between collections of pairs described in Section 4.1.4 and by visualizing the latent space representation learned. The second question is answered by analyzing the extracted topics evaluated through coherence measure and by assessing the quality of the learned representations using clustering metrics, also described in Section 4.1.4.

4.2.1 Similarity between Clinical Terms

As mentioned previously the performance of each model was evaluated from three perspectives, namely cosine similarity between pairs of extracted terms, computing the consistency of the the topics generated by each model through topic coherence scores, and based on clustering performance. Better performing models will have a higher topic coherence, as well as being be able to produce higher cosine similarity for pairs of codes from the same ICD-10 or ICPC-2 chapter, while keeping a low cosine similarity for pairs from distinct chapters (i.e., the difference between same chapter and distinct chapter pairs should be more accentuated) as well as producing clearly defined clusters (i.e., in the perfect case each cluster should only contain data points of a single chapter) computed through both supervised and unsupervised clustering metrics introduced previously.

The results for the similarity results for code-pairs and phrase-pairs collections are displayed in Ta-
Table 4.5: Average cosine similarity between pairs of ICD-10 and ICPC-2 codes considering 25 latent components with respective difference between pairs.

<table>
<thead>
<tr>
<th>Model</th>
<th>ICD</th>
<th>ICPC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SC</td>
<td>DC</td>
</tr>
<tr>
<td>FR-NMF</td>
<td>0.342</td>
<td>0.170</td>
</tr>
<tr>
<td>KL-NMF</td>
<td>0.220</td>
<td>0.059</td>
</tr>
<tr>
<td>SNMF</td>
<td>0.866</td>
<td>0.811</td>
</tr>
<tr>
<td>DSNMF</td>
<td>0.753</td>
<td>0.754</td>
</tr>
<tr>
<td>AE</td>
<td>0.779</td>
<td>0.772</td>
</tr>
</tbody>
</table>

Table 4.6: Average cosine similarity between pairs of ICD-10 and ICPC-2 phrases considering 25 latent components with respective difference between pairs.

<table>
<thead>
<tr>
<th>Model</th>
<th>P ICD</th>
<th>P ICPC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SC</td>
<td>DC</td>
</tr>
<tr>
<td>FR-NMF</td>
<td>0.243</td>
<td>0.130</td>
</tr>
<tr>
<td>KL-NMF</td>
<td>0.102</td>
<td>0.031</td>
</tr>
<tr>
<td>SNMF</td>
<td>0.821</td>
<td>0.796</td>
</tr>
<tr>
<td>DSNMF</td>
<td>0.765</td>
<td>0.756</td>
</tr>
<tr>
<td>AE</td>
<td>0.791</td>
<td>0.785</td>
</tr>
</tbody>
</table>

From all methods, the SNMF achieved highest cosine similarity (i.e., skewed towards the value of 1), however this result is common across pairs from same chapter and different chapter. Similarly, the DSNMF also inherits the same problems of SNMF, although the values are less skewed towards 1. These results are justified by poor initialization techniques employed by these methods, contrary to NMF-based techniques which use NNDSVDa initialization allowing to create latent representations that better differentiate codes from same chapter or distinct chapter. The best performing methods are not those with higher cosine similarity overall, but those methods that show a higher similarity difference between pairs in the same chapter or distinct chapter (SC - DC), and particular to ICD-10, also higher difference between pairs within the same chapter and block (SB) than within the same chapter (SC) or from distinct blocks (DB), i.e., the best methods create higher differences SB - SC and SB - DB.

Taking the previous statement into consideration, the best performing methods are the FR-NMF and the KL-NMF, which use the Frobenius norm and the Kullback-Leibler divergence, respectively as described in Section 2.2.1. Since the difference between cosine similarity of ICD SB - SC and ICD SC - DC given by these methods is higher than any other method, indicating that both approaches based on NMF are better at differentiating between codes from same chapters and different chapter, and by analogy better at preserving the hierarchical structure of the ICD-10 classification system. Furthermore, both these methods produced slightly higher cosine similarity for code pairs from the same ICPC-2 chapter (ICPC SC), although the difference (ICPC SC - DC) is not statistically significant as it is demonstrated by the box plots in Figure 4.6 because the notches of ICPC SC and ICPC DC overlap, hence we are left unable to conclude if the ICPC-2 hierarchical structure is preserved by any method based on collections of code-pairs.
Now, let us analyze the results achieved for phrase-pairs collection (Table 4.6), which for the most part are similar to the results of code-pairs collection for the ICD-10 classification system, i.e., codes from the same chapter and block (P ICD SB) produced higher similarity between them than pairs from same ICD-10 chapter (P ICD SC), and these scored higher than pairs from different chapter (P ICD DC) or different block (P ICD DB). For the ICPC-2 classification, there was an improvements on differentiating between same chapter (P ICPC SC) and distinct chapter (P ICPC DC) for FR-NMF and KL-NMF methods. In this case, we can conclude the difference P ICPC SC - DC is statistically significant as the notches of P ICPC SC and P ICPC DC do not overlap as portrayed in Figure 4.7. Meanwhile, for SNMF, DSNMF and AE models there was no improvement, producing the nearly the results as when considering collection of code-pairs in Table 4.5.

To better understand the variability within and between collections created achieved by FR-NMF model, we present the cosine similarity for code-pairs and phrase-pairs collection in Figure 4.6 and Figure 4.7, these figures are also used to test if difference between two collections is statistically significant, in case the notches of two different box plots do not overlap. Hence, for the FR-NMF model only the difference between ICPC SC - DC is not statistically significant.

The results achieved let us conclude that only NMF based methods (FR-NMF and KL-NMF) are able to capture the inter-chapter and intra-chapter differences for both classification systems, i.e., these preserved the hierarchical organization in chapters for ICPC-2 or in chapters and blocks for ICD-10, when considering code-pairs and phrase-pairs as seen by the difference in Table 4.5 and Table 4.6.

The evaluation previously described is an indirect measure of the quality of the representations.
learned, by assessing the hierarchical structure of a medical classification system based on similarity of pairs. However, in the biomedical domain, there is a need to analyze the results achieved more directly, for instance, to assess the relationship between clinical terms extracted by an expert in the domain. Thus, one of most important aspects of this work is the interpretability of the results, which is a critical factor in the medical domain, and the reason behind the employment of NMF and variants techniques in our experiments, as these techniques leverage non-negative representations. Hence, in order to assess the results of each method, we use similarity visualizations through clustermap (see Figure 4.8) and show a table of the generated topics (see Table 4.7), where each topic is a dimension of the final latent representation. By employing these techniques allows to directly inspect the results obtained, thus being more interpretable and more useful for medical staff which can use these to assess the significance of the results.

To obtain the clustermap similarity visualization, one must first select which are the clinical terms or codes to compare and visualize. For instance, in Figure 4.8, we started by selecting a subset of ICD-10 all codes extracted by the FR-NMF model, starting from chapter 13 up to 20. However, there are no codes from chapters 15, 16 and 17, as we sampled a subset $D_S^i$ from all documents and there are a few number of documents assigned with these chapters (see Figure 4.2). The criteria for selecting codes between the chapter 13 and 20 was arbitrary in this case, however the architecture allows to select any code or clinical term for comparison, as long as these were considered a feature by the model. One of the benefits of this approach is that it allows a medical expert to cherry-pick specific codes or clinical terms to compare, when there is a strong evidence or background knowledge that these might
Figure 4.8: Cosine similarity between codes extracted by FR-NMF with latent representation size of $\tau = 5$, taking into consideration all blocks from chapters 13, 14, 18, 19 and 20 of the ICD-10 classification system. Showing the strong correlation between accidental falls (W199) and renal insufficiency complications (N10, N189, N390).

be related, thus allowing to verify if that is the case or if the model was capable of capturing known relationships.

After the codes or clinical terms have been selected for comparison, we compute the cosine similarity between a code or a clinical term against every other one, yielding a similarity matrix $S$, where each cell $S[i, j]$ encodes similarity between a code or clinical term in row $i$ with another code or clinical term in column $j$. Hence, the matrix $S$ allows one to easily spot the similarities between two or more codes, codes against clinical terms, or clinical terms against other clinical terms. The cell value is determined by the similarity metric used, in our case we employed cosine similarity, thus every cell in $S$ encodes a high similarity when the value is close to 1, and a value of 0 indicates no similarity.
Figure 4.8 shows the cosine similarity between the extracted codes from different chapters, producing interesting relationships. For instance, there is a strong relationship between accidental falls (code W199) and renal insufficiency complications encoded with (N10, N19, N179, N189). These findings are supported in the medical literature as reported by López-Soto et al. [2015] and Papakonstantinopoulou and Sofianos [2017]. Both teams showed that elderly patients of chronic kidney diseases are more prone to injuries, and with higher probability of fall re-incidence. The authors show that other risk factors might be at play including malnutrition, Vitamin-D deficiency and more common occurring in hemodialysis patients. Although these findings show high correlation, López-Soto et al. [2015] argue that more research is needed in order to precisely calculate the incidence and risk factors of falls in population with chronic insufficiency in order to take preventive measures that you reduce the risk of fatal accidental falls in this population.

Moreover, the results show that FR-NMF model is able to differentiate between clearly defined causes of death, such as car accidents (V892), against other codes which encode health complication or natural causes, only showing similarity with unspecified intracranial injury (S069), which might be related to the fact that these are used in combination by a healthcare professional when assigning the codes for the cause of death.

### 4.2.2 Quality of the Learned Representations

This section reveals the results produced by all methods in terms of topic modelling and clustering evaluation measures. Based on these, we are able to assess the quality of the representations by changing by the $\tau$ parameter which controls the dimension of the latent representation.

**Topic coherence**

Every method tested can be analyzed from a topic modelling perspective, where the extracted keyphrase is represented as column in the latent representation, i.e., the $H \in \mathbb{R}_+^{\tau \times v}$ matrix having $\tau$ as the size of the representation and $v$ is the number of features, in our case each feature is phrase which can be a clinical term, a code or any other word that the model considers relevant. The matrix $H$ encodes the significance of a feature column to a specific topic (row) with dimension given by $\tau$. For instance, to obtain the most important features in a topic, we sort every row by significance for all features $v$, and after select only the $n$ most significant features for each row $\tau$. Considering the representation $H$ created by FR-NMF model, setting $\tau = 5$ and selecting the 10 top most important terms (i.e., $n = 10$), we generate the topics displayed in Table 4.7. This topic table allows for human inspection and assessment if the most important features (clinical terms) obtained are consistent, related and refer to only one topic, or if the features in a topic are unrelated.

For instance, the Topic #02 refers to cerebrovascular accident and arterial hypertension conditions, in this topic the most important features are the tokens that compose these conditions, namely cerebral, vascular, acidente, arterial, and hipertensão, as well as the complete name of the condition acidente vascular cerebral and hipertensão arterial, and the ICD-10 codes i64 (Stroke, not specified as haem-
Figure 4.9 shows the CV coherence scores across multiple models implemented. The coherence results obtained align the cosine similarity results, having both methods based on NMF achieve higher coherence scores across all tested topic dimensionalities parametrized by $\tau$.

**Clustering performance**

As mentioned previously in Section 4.1.4, the third perspective that we evaluate the methods employed is based on the clustering performance achieved by each. To do so, we use supervised (NMI and Homogeneity) and unsupervised (DBI and Silhouette coefficient) metrics. To estimate these, we employed $k$-means clustering algorithm over the intermediate representations (TF-IDF and Doc2Vec), as well as over the final latent representations for each model parametrized by $\tau \in \{5, 10, 15, 20, 25\}$, then took the average to obtain an aggregated result. One must notice that the dimensionality of the
intermediate representation was fixed, only the final representation was parametrized by $\tau$. To measure the clustering performance we did not use all values $\tau$, that were considered to calculate the topic coherence, because we saw that increasing the dimensionality yields worse results. Moreover, by testing with multiple values of $\tau$ avoids poor initializations which lead to local minima. The $k$-means clustering is parametrized by the number of clusters $k$, in our case we use the number of distinct ICD-10 chapters, and in our particular dataset sample ($DS_1$) there were a total of 16 (i.e., $k = 16$) chapters out of 20. The chapters not included were chapter 7, 8, 15 and 16 as these have a low representation in the dataset, as shown in Figure 4.2.

The results achieved by different representations are displayed in Table 4.8, also including the results of the intermediate representation through TF-IDF and Doc2vec, in order to assess if the final representation improved on metrics than intermediate ones. All methods employed improved the results both for unsupervised and supervised metrics, with the exception of SNMF and DSNMF in supervised case, as these two methods scored lower after transforming the TF-IDF representation. The best results in terms of clustering performance was achieved by AE followed by NMF for supervised metrics and by DSNMF in unsupervised. More specifically, the combination TF-IDF as intermediate representation transformed by AE achieved best results in supervised metrics, while the best results in unsupervised was also produced by AE although in this case having Doc2Vec as intermediate representation.

Furthermore, all methods produced similar results for NMI and Homogeneity when using the Doc2Vec representation, although there was a small improvement over the initial representation. We suspect this is related to the fact that the initial size of the paragraph vector chosen was 2000, hence leading to sparse intermediate representations.

After these results we can conclude that the neural network based methods (DSNMF and AE) performed better in unsupervised metrics, creating more well defined clusters than NMF based methods, indicating that the latter tend to create overlapping clusters. On the other hand, FR-NMF and KL-NMF produced better results in supervised for either TF-IDF or Doc2Vec as intermediate representations, aligning with the coherence scores achieved in topic modelling.
<table>
<thead>
<tr>
<th>Representation</th>
<th>Supervised</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NMI</td>
<td>Homogeneity</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>0.202</td>
<td>0.141</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>0.013</td>
<td>0.007</td>
</tr>
<tr>
<td>TF-IDF + FR-NMF</td>
<td>0.214</td>
<td>0.150</td>
</tr>
<tr>
<td>TF-IDF + KL-NMF</td>
<td>0.213</td>
<td>0.150</td>
</tr>
<tr>
<td>TF-IDF + SNMF</td>
<td>0.196</td>
<td>0.140</td>
</tr>
<tr>
<td>TF-IDF + DSNMF</td>
<td>0.070</td>
<td>0.049</td>
</tr>
<tr>
<td>TF-IDF + AE</td>
<td><strong>0.215</strong></td>
<td><strong>0.152</strong></td>
</tr>
<tr>
<td>Doc2Vec + NMF</td>
<td>0.033</td>
<td>0.024</td>
</tr>
<tr>
<td>Doc2Vec + KL-NMF</td>
<td>0.033</td>
<td>0.025</td>
</tr>
<tr>
<td>Doc2Vec + SNMF</td>
<td>0.033</td>
<td>0.024</td>
</tr>
<tr>
<td>Doc2Vec + DSNMF</td>
<td>0.033</td>
<td>0.024</td>
</tr>
<tr>
<td>Doc2Vec + AE</td>
<td>0.032</td>
<td>0.023</td>
</tr>
</tbody>
</table>

Table 4.8: Average $k$-means clustering between all representations considering multiple different dimensionalities $\tau \in \{5, 10, 15, 20, 25\}$ for the latent representation.
5.1 Contributions

The main contributions in this work was the creation of an architecture that uses key-extraction techniques such as AutoPhrase, vector space representations (TF-IDF and Doc2Vec) and dimensionality reduction techniques (NMF, SNMF, DSNMF and AE) that processes a dataset from the medical domain extracting relevant clinical terms present in the data. After extraction and representation we compared each dimensionality technique based on the ability to preserve the hierarchical structure (both intra-chapter and inter-chapter) of two classification system, namely ICD-10 and ICPC-2, showing that only methods based on NMF were capable of preserving the hierarchical structure.

Furthermore, we also measured the performance of multiple dimensionality reduction techniques from a topic modelling and clustering standpoint. We showed that from a topic modelling perspective the neural networks based representation learning/dimensionality reduction techniques, such as DSNMF and AE, showed worse results than methods based on NMF. These results are related to the fact that neural network based techniques have been proposed for clustering and reconstruction tasks, in which we also tested, and indeed showed better performance than NMF based methods.

The main limitations of this work it that the performance of the models have yet to be tested under a different dataset in order to assess how these perform under different conditions. For instance, use other dataset encoded with the ICD-9 classification, or use one that contains more extensive descriptions, as the one used only contains concise descriptions with majority of documents containing fewer than 12 words. Hence, validate if under such conditions the NMF based methods perform better than deep neural networks based techniques. Furthermore, the poor results achieved when comparing the similarity between code-pairs and phrase-pairs in the ICPC-2 classification system is related to the fact that the there were only mapped 8 452 documents with a high imbalance in terms of codes and chapters distribution, which meant that the number of distinct codes and possible combination of pairs was low when compared to ICD-10.

Another limitation is the fact that the topic generated and codes and phrase relationships found have yet to be validated by a medical expert. His expertise would allow to obtain more insight if the terms on a
given topics are correlated, i.e., each topic generated is coherent to the medical knowledge. Moreover, his vast knowledge in the field would also allow to select and validate hand-picked codes or phrases, to analyze and visualize their relationship through a similarity matrix.

5.2 Future Work

For future work we seek to explore unsupervised key-extraction techniques such as YAKE, instead of AutoPhrase, proposed by Campos et al. [2019] and run experiments with other dimensionality reduction methods based on autoencoder extensions which are more closely related to NMF and to topic modelling, e.g., PAE-NMF proposed by Squires et al. [2019] or NVDM proposed by Miao et al. [2016] which were not considered in this work, as these methods can not be compared based on the cosine similarity, because these produce a probability distribution in the latent space and not $n$-dimensional vector for a code or phrase representation. Hence, the evaluation metrics used would also need to be changed in order to compare distribution, for instance by using Kullback-Leibler divergence instead of cosine similarity between two ICD-10 or ICPC-2 codes vector representations.

More research is need in order to adapt DSNMF and AE to produce better in text processing tasks, as these methods have initially been proposed for image processing. For instance, one of the adaptations would be to use an attention layers on the AE. This can be achieved by using a recurrent neural network based on long-short term memory (LSTM), which has successively shown state of the art performance in text processing tasks, because the LSTM network is more capable of capturing the relationship between terms than the current feedforward network which is used in these methods.


Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. Mining Quality Phrases from Massive Text Corpora. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1729–1744, 2015.


Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. Automated Phrase Mining from Massive Text Corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30:1825–1837, 2018.


Appendix A

Code and Clinical Terms Similarity

In Figure A.1-A.6 are visualized the similarity matrix between code and phrases partitioned by the ICD-10 or ICPC-2 chapter obtained by the FR-NMF. The visualization of similarity between between codes from chapters 13 to 20 of ICD-10, is not included as it was already presented in Figure 4.8 in Section 4.2.

This appendix also shows the similarity matrix between the top 30 codes for all models employed in the architecture, starting from Figure A.7 to Figure A.11. One can notice that in Figure A.8 which shows the similarity matrix between codes extracted by KL-NMF creates higher similarity differences between related and unrelated terms than any other method. Furthermore, also possible to visualize that the similarities cells in Figure A.9 (SNMF), Figure A.10 (DSNMF) and in Figure A.11 (AE) are overall skewed toward a value of 1, which leads to poor differentiation between codes from same chapter and those from a distinct chapter, as demonstrated in Table 4.5.

All similarity matrices visualized were created by choosing a value for the latent dimension of $\tau = 5$, as this value has shown to produce good topic coherence results across all methods tested.
Figure A.1: Similarity between codes from chapters 1 to 6 of ICD-10 by FR-NMF.
Figure A.2: Similarity between codes from chapters from 7 to 12 of ICD-10 by FR-NMF model.
Figure A.3: Similarity between clinical terms encoded with codes from chapters 1 to 6 of ICD-10 by FR-NMF.
Figure A.4: Similarity between clinical terms encoded with codes from chapters 7 to 10 of ICD-10 by FR-NMF.
Figure A.5: Similarity between clinical terms encoded with codes from chapters 11 to 20 of ICD-10 by FR-NMF.
Figure A.6: Similarity between clinical terms coded with the ICPC-2 classification by FR-NMF.
Figure A.7: Similarity between 30 most common ICD-10 codes in FR-NMF model.
Figure A.8: Similarity between 30 most common ICD-10 codes in KL-NMF model.
Figure A.9: Similarity between 30 most common ICD-10 codes in SNMF model.
Figure A.10: Similarity between 30 most common ICD-10 codes in DSNMF model.
Figure A.11: Similarity between 30 most common ICD-10 codes in AE model.
Follows the topic tables generated by each method employed in our experiments. These tables were generated by sorting the important feature in each latent dimension. The following tables all were created by choosing a value for the latent dimension of $\tau = 5$, as this value has shown to produce good topic coherence results across all methods tested.

The results achieved by the FR-NMF model (Table B.1) were already discussed in Section 4.2.2. Similar results were also produced by KL-NMF and SNMF in Table B.2 and Table B.3, respectively. Although the SNMF model was not able to capture the topic that aggregates pneumonia related terms, which was captured in Topic #05 by FR-NMF and KL-NMF models. Meanwhile, the topics produced by DSNMF are much more incoherent, since it aggregates unrelated codes (i350), clinical terms (*neoplasia laringe*) and other words (*totalmente dependente, devida*) into the same topic, as portrayed by Topic #01 in Table B.4. Interestingly, the DSNMF model aggregated non-clinical terms, i.e., descriptive words (*encontrada, altura, sinais, manobras suporte*) used by healthcare professionals when writing the cause of death, as shown in Topic #03 in Table B.4. The AE model produced topics that contain the same clinical terms in several of them. For instance, the clinical term *doença pulmonar obstrutiva cronica* and its sub-parts (i.e., *doença pulmonar, pneumonar cronica, obstrutiva cronica*) appear in topics #03, #04 and #05, thus this model shows worse performance in terms discrimination of the distinct topics.

<table>
<thead>
<tr>
<th>Topic # 01</th>
<th>Topic # 02</th>
<th>Topic # 03</th>
<th>Topic # 04</th>
<th>Topic # 05</th>
</tr>
</thead>
<tbody>
<tr>
<td>insuficiência</td>
<td>cerebral</td>
<td>neoplasia</td>
<td>insuficiência</td>
<td>pneumonia</td>
</tr>
<tr>
<td>respiratória</td>
<td>vascular</td>
<td>carcinoma</td>
<td>respiratória</td>
<td>j189</td>
</tr>
<tr>
<td>doença</td>
<td>acidente vascular</td>
<td>pulmão</td>
<td>cardíaca</td>
<td>pneumonia aspasão</td>
</tr>
<tr>
<td>crónica</td>
<td>acidente</td>
<td>metastização</td>
<td>insuficiência cardíaca</td>
<td>bilateral</td>
</tr>
<tr>
<td>insuficiência respiratória</td>
<td>vascular cerebral</td>
<td>adenocarcinoma</td>
<td>insuficiência respiratória</td>
<td>pneumonia bilateral</td>
</tr>
<tr>
<td>cardíaca</td>
<td>acidente vascular cerebral</td>
<td>metastizado c349</td>
<td>insuficiência cardíaca</td>
<td>sepis</td>
</tr>
<tr>
<td>diabetes</td>
<td>i64</td>
<td>metastizada</td>
<td>insuficiência renal</td>
<td>j189 pneumonia</td>
</tr>
<tr>
<td>renal</td>
<td>acidente vascular</td>
<td>maligna</td>
<td>renal</td>
<td>nosocomial</td>
</tr>
<tr>
<td>insuficiência cardiaca</td>
<td>arterial</td>
<td>neoplasia maligna</td>
<td>cronica</td>
<td>associada</td>
</tr>
</tbody>
</table>

Table B.1: Topics generated by FR-NMF model.
Table B.2: Topics generated by KL-NMF model.

<table>
<thead>
<tr>
<th>Topic # 01</th>
<th>Topic # 02</th>
<th>Topic # 03</th>
<th>Topic # 04</th>
<th>Topic # 05</th>
</tr>
</thead>
<tbody>
<tr>
<td>insuficiência cerebral</td>
<td>neoplasia</td>
<td>neoplasia</td>
<td>insuficiência cardiaca</td>
<td>pneumonia</td>
</tr>
<tr>
<td>respiratória vascular</td>
<td>adenocarcinoma</td>
<td>respiratória</td>
<td>j189</td>
<td>sepsis</td>
</tr>
<tr>
<td>câncer</td>
<td>pulmão</td>
<td>pulmão</td>
<td>insuficiência respiratória</td>
<td>choque</td>
</tr>
<tr>
<td>doença vascular cerebral</td>
<td>adenocarcinoma</td>
<td>adenocarcinoma</td>
<td>insuficiência respiratória</td>
<td>sepsis</td>
</tr>
<tr>
<td>cardíaca</td>
<td>metástase</td>
<td>insuficiência</td>
<td>insuficiência respiratória</td>
<td>aspiração</td>
</tr>
<tr>
<td>pulmonar</td>
<td>vascular</td>
<td>respiratória</td>
<td>insuficiência cardiaca</td>
<td>sépsis</td>
</tr>
<tr>
<td>cardiopatia renal</td>
<td>insuficiência cardiovascular</td>
<td>insuficiência respiratória</td>
<td>demência</td>
<td>nosocomial</td>
</tr>
<tr>
<td>agudo</td>
<td>insuficiência</td>
<td>respiratória</td>
<td>nosocomial</td>
<td>bilateral</td>
</tr>
</tbody>
</table>

Table B.3: Topics generated by SNMF model.

<table>
<thead>
<tr>
<th>Topic # 01</th>
<th>Topic # 02</th>
<th>Topic # 03</th>
<th>Topic # 04</th>
<th>Topic # 05</th>
</tr>
</thead>
<tbody>
<tr>
<td>congestiva</td>
<td>choque septico</td>
<td>encontrado</td>
<td>hipercalemia</td>
<td>submicrórdio</td>
</tr>
<tr>
<td>diabetes</td>
<td>hipertensão arterial fístula</td>
<td>choque hipovolêmico</td>
<td>crônica anemia</td>
<td>i678</td>
</tr>
<tr>
<td>insuficiência cardiaca</td>
<td>insuficiência respiratória pneumonia</td>
<td>altura</td>
<td>base</td>
<td>saos</td>
</tr>
<tr>
<td>insuficiência respiratória</td>
<td>insuficiência respiratória</td>
<td>sinais</td>
<td>alteração</td>
<td>lesões traumáticas</td>
</tr>
<tr>
<td>insuficiência cardiaca</td>
<td>adenocarcinoma</td>
<td>aldeíla de adquirida</td>
<td>anemia crônica</td>
<td>acamada totalmente dependente</td>
</tr>
<tr>
<td>metastizada</td>
<td>vascular</td>
<td>pancreatite</td>
<td>fractura</td>
<td>y349</td>
</tr>
<tr>
<td>insuficiência</td>
<td>insuficiência</td>
<td>hta dislipidemia</td>
<td>cirrhose hepática</td>
<td>acidente vascular</td>
</tr>
<tr>
<td>insuficiência</td>
<td>insuficiência</td>
<td>crise</td>
<td>hipertensão arterial</td>
<td>insuficiência cardiaca</td>
</tr>
<tr>
<td>insuficiência</td>
<td>insuficiência</td>
<td>manobras suporte</td>
<td>urgencia</td>
<td>derrame pleural</td>
</tr>
</tbody>
</table>

Table B.4: Topics generated by DSNMF model.

<table>
<thead>
<tr>
<th>Topic # 01</th>
<th>Topic # 02</th>
<th>Topic # 03</th>
<th>Topic # 04</th>
<th>Topic # 05</th>
</tr>
</thead>
<tbody>
<tr>
<td>acidente vascular cerebral</td>
<td>pulmão</td>
<td>pulmão</td>
<td>doença pulmonar</td>
<td>pulmão</td>
</tr>
<tr>
<td>acidente vascular cerebral</td>
<td>i64 acidente vascular</td>
<td>pulmão obstrutiva</td>
<td>doença pulmonar</td>
<td>pulmão</td>
</tr>
<tr>
<td>vascular cerebral</td>
<td>enforte agudo miocárdio</td>
<td>obstrutiva crônica</td>
<td>doença pulmonar</td>
<td>pulmão</td>
</tr>
<tr>
<td>hipertensão arterial diabetes</td>
<td>carcinoma</td>
<td>pulmão obstrutiva</td>
<td>pulmão obstrutiva</td>
<td>pulmão</td>
</tr>
<tr>
<td>acidente vascular</td>
<td>i64 acidente micardio</td>
<td>pulmão obstrutiva</td>
<td>pulmão obstrutiva</td>
<td>pulmão</td>
</tr>
<tr>
<td>acidente</td>
<td>neoplasia pulmão</td>
<td>pulmão obstrutiva</td>
<td>pulmão obstrutiva</td>
<td>pulmão</td>
</tr>
<tr>
<td>miocárdio</td>
<td>agudo miocárdio</td>
<td>obstrutiva crônica</td>
<td>pulmão obstrutiva</td>
<td>pulmão</td>
</tr>
<tr>
<td>insuficiência</td>
<td>desconhecido</td>
<td>enfermeiro agudo</td>
<td>doença pulmornar</td>
<td>pulmão</td>
</tr>
<tr>
<td>fístula</td>
<td>cancro</td>
<td>pulmão obstrutiva</td>
<td>pulmão obstrutiva</td>
<td>pulmão</td>
</tr>
<tr>
<td>acidente vascular</td>
<td>pulmão obstrutiva crônica</td>
<td>pulmão obstrutiva</td>
<td>pulmão obstrutiva</td>
<td>pulmão</td>
</tr>
<tr>
<td>enfarte agudo miocárdio</td>
<td>pulmão obstrutiva crônica</td>
<td>pulmão obstrutiva</td>
<td>pulmão obstrutiva</td>
<td>pulmão</td>
</tr>
</tbody>
</table>

Table B.5: Topics generated by AE model.